Representation and Measure of Structural Information

Hiroshi Ishikawa

Department of Information and Biological Sciences

Nagoya City University

Nagoya 467-8501, Japan

hi@nsc.nagoya-cu.ac.jp

Abstract

We introduce a uniform representation of general objects that captures the regularities with respect to their structure. It allows a representation of a general class of objects including geometric patterns and images in a sparse, modular, hierarchical, and recursive manner. The representation can exploit any computable regularity in objects to compactly describe them, while also being capable of representing random objects as raw data. A set of rules uniformly dictates the interpretation of the representation into raw signal, which makes it possible to ask what pattern a given raw signal contains. Also, it allows simple separation of the information that we wish to ignore from that which we measure, by using a set of maps to delineate the *a priori* parts of the objects, leaving only the information in the structure.

Using the representation, we introduce a measure of information in general objects relative to structures defined by the set of maps. We point out that the common prescription of encoding objects by strings to use Kolmogorov complexity is meaningless when, as often is the case, the encoding is not specified in any way other than that it exists. Noting this, we define the measure directly in terms of the structures of the spaces in which the objects reside. As a result, the measure is defined relative to a set of maps that characterize the structures. Though it does not depend on Kolmogorov complexity, it raises a question of their relationship, as the class of applicable objects includes strings. It turns out that the measure is equivalent to Kolmogorov complexity when it is defined relative to the maps characterizing the structure of natural numbers. Thus, the formulation gives the larger class of objects a meaningful measure of information that generalizes Kolmogorov complexity.

1   Introduction

What is a pattern? There does not seem to be a generally accepted mathematical definition. Intuitively, a pattern is something simpler than it is apparent. For instance, a repetition of

a short substring in a longer string is a pattern: the longer string is simpler, or contains less information, than most other strings of the same length. Here, we see a comparison between the apparent size (length in the literal representation) and the "real" amount of information. Formally, this can be stated in terms of Kolmogorov complexity[1, 2, 9, 13, 14] of the string, which is roughly defined as the length of the shortest input to a universal Turing machine that produces the string. A string can be said to have a pattern if its Kolmogorov complexity is much smaller than its length: strings that can be effectively described in a significantly shorter description than their length have patterns. Our goal in this paper is to formalize this notion in the domain of more general objects than strings.

For instance, consider bitmap images. Ordinary images are much orderly than what is allowed by their representation as an array of colors; if we take a random bitmap out of all that can be represented as a bitmap, it is almost always a white noise, rather than what we would consider an ordinary image. This is similar to the string case where most strings of a given length are random ones that do not have a shorter description than the literal one. What is the corresponding "effective description" of images? Intuitively, it should be a way to describe the image in which ordinary images can be represented more concisely than noise images.

A Turing machine that produces a bitmap does not suffice because, unlike the case of strings, where all strings can be represented precisely as they are, the bitmaps are only approximations of what we consider to be real images: Pixels are artifacts of arbitrary approximation; and we naturally consider bitmaps of various resolutions as the same, if they show the same scene. There would be no problem if it were the case that all important features of an image are independent of the choice of pixelation. However, this is clearly not so: even a notion as simple as that of a line is not so simple to define on bitmaps, especially in such a way a line in one resolution can be converted into a line in another resolution.

Infinite resolution bitmaps, or functions on an image domain that takes values in the color space, seem to be good enough for the literal representation. But then, the objects appearing there are continuous, infinite entities and thus cannot easily be described effectively as, for instance, an output of a Turing machine. Yet intuition tells us that some of these infinite entities contain only finite information, as the extreme cases of "geometric" visual patterns shown in Figure 1.

1.1 Kolmogorov Complexity Covers All?

But surely, one might say, Kolmogorov complexity already covers any domain, since Computer Science teaches us that information can be encoded by strings. That is, we can first fix some standard enumeration of the objects, establishing a one-to-one correspondence between the objects and strings; then we can define the complexity of an object to be the Kolmogorov complexity of the corresponding string. That seems to be where such an inquiry usually stops, content with the notion that essentially we only need to investigate
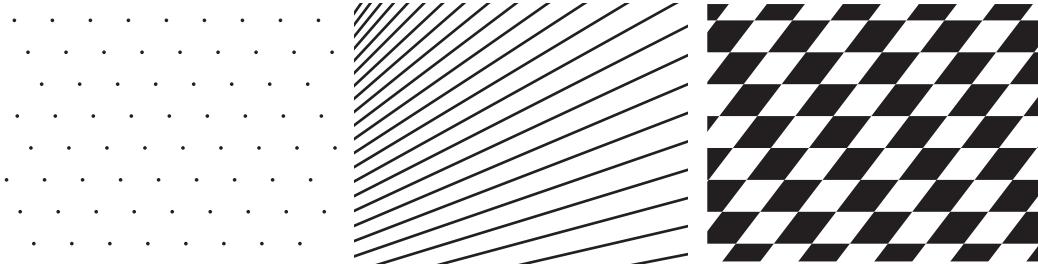
Figure 1: Visual patterns.

strings.

However, we immediately encounter a few problems.

First, for a class of objects (such as subsets of a Euclidean space) that has a larger cardinality than the set of all strings, we cannot encode all objects by strings; thus we need to give up the one-to-one correspondence. We must either encode only some of the objects, encode (perhaps an infinite number of) multiple objects by each string, or employ some combination of the two approaches. The choice amounts to knowing what to ignore, whether it is some (even most) of the objects that are not encoded, or the difference between the objects that are encoded into the same string. *How should we make this choice?*

More fundamentally, the resulting measure has little meaning without actually specifying the encoding. Let O and S be the sets of objects and strings, respectively, and $K(s)$ the Kolmogorov complexity of a string s. With an encoding $f : O \to S$, we might call $K(f(x))$ the complexity of object x. However, if we do not have some good reason to take a particular f, we can equally use the encoding $p \circ f$ with an arbitrary permutation $p : S \to S$. This observation renders the definition meaningless without f explicitly specified. So the question is: *what is the encoding that gives the complexity some meaning?* How can we avoid falling into this trap of arbitrariness? It is certainly not enough just to say that it can be encoded.

With strings, we can choose the identity map as f, which gives $K(f(x))$ as much meaning as $K(s)$. In other cases, however, we need to specify f, with at least some justification. If we insist encoding objects into strings, we need to define a concrete encoding for each class of objects.

Another problem of measuring the information solely through Kolmogorov complexity is that we cannot easily ignore the part of information we do not care. For instance, we may try to represent a point in the Euclidean plane by identifying the space with $\mathbb{R}^2$, i.e., by a pair of real coordinates, and then encoding them by strings. However, a single real number can contain an arbitrarily large amount of information. Thus, in this representation, a single point can have an arbitrarily large information when encoded by a string. That is certainly not what we want here. Thus, an important part of the encoding is specifying the part of the information we wish to ignore. But we cannot simply delete such information

in the encoding process, since it may be needed to identify and measure the regularities in the structure later. If we insist that the computation be carried out strictly in the domain of strings, f's output must contain all the information in the points. But after the information has been converted into strings, how do we specify which part of the information should be ignored?

It would be much better if we can define the notion of computation, such as compression and pattern finding, directly in terms of the objects we deal with. What we offer in this paper is a meta definition of f for multiple classes of objects by specifying how to embed computations in larger spaces. Central to the formalism is the representation of objects that offers the means to specify the information in individual elements that should be ignored, while using that very information to find the structures, in which we wish to measure the amount of information.

Thus, the central problem is that of encoding, or representation, of objects. The paradigm to measure information through computation is the same; the difference is where the computation takes place. This question of encoding seems to have suffered a neglect which, in our belief, has prevented us from formulating a notion of information in objects that have not already been encoded in a convenient way. We discuss this further in Section 7.

1.2 Motivation

Our motivation for asking this question stems from the desire to model perception. For perception, there needs to be a large amount of prior knowledge stored in the perceiver, because perception is an inherently ill-posed problem. Perception is a process in which the configuration of the signal source is recovered from a signal, as in recovering a three-dimensional scene from an image.

The problem is that, given the signal, there are usually infinitely many possible source configurations. Without a preference of possible source configuration on the side of the perceiver, there is no reason to choose one possibility over another. For instance, our visual system has a great capability to organize the visual signal into interpretable shapes, like making sense of the famous Dalmatian photo by R. C. James in [5]. To model such a system, it is not enough to know what the possible configurations of the signal source are; we need to know in advance how likely we are going to encounter each of them.

However, even putting aside the problem of estimating the probabilities, just storing and retrieving the data is impossible unless we have a very good way to compress the data; for instance, if we store the possible shape of surfaces as an array of 10 possible heights at each of 10×10 positions, the number of possible surfaces would be 10^{100}. The way this problem has been dealt with is by estimating the probabilities by looking at specific characteristics of the possible surface. For instance, the surface smoothness can be computed from a given description of the surface; we can then decide, for instance, that the smoother the surface is, the higher the probability. Indeed, the area of computer vision and pattern recognition is full of such heuristics. Even when machine learning techniques are used, the variables to

be learned must be carefully chosen because we cannot simply learn all possible surfaces.

Our desire to have a measure of information originates from the wish to have a principle for automatically deciding which quantity to look at and which combinations of variables to learn, because we consider it reasonable for a perceiving entity to look first for simpler patterns in the signal, as well as because of the demand of storage efficiency. That is, if we have a measure of simplicity of general visual objects, we can say, for instance, that the probability is proportional to the simplicity, or use machine learning techniques to learn the probabilities that such simpler patterns appear.

In more general terms, this is a problem of inductive inference and modeling: we inductively seek a model of the world that best explains the data. There are theories that treat such a problem, and among them are ones with the spirit we describe above. For instance, the Minimum Description Length (MDL) principle[11] advocates Occam's Razor. Among models that equally fit the data, it chooses the one that is the "simplest" in the sense that it allows for a shorter description of the data. However, crucially missing from this theory is the problem of representation. The MDL theory only deals with strings as the data and does not say how the objects should be described by strings. It is a good principle for people trying to deal with individual problems they understand; but when it comes to dealing with general objects, it lacks the mathematical concreteness needed to program the principle itself into machines.

Also, as perceiving entities, we seem to have more interest in the finite part of the data. One may even say that we can only perceive the finite information out of any infinitely rich source of information, on the basis that our capacity of representation is presumably finite. For instance, if we see a white noise image, we do not perceive the amount of information that can be encoded in such an image. Instead, we glean the information that we can; we might just note that it is a white noise, or if it is a video we would recognize that the noise is constantly changing, and so on. If we see the three images in Figure 2, which are the same pattern with different noise added, we do not discriminate among them. Even though as raw bitmaps they are quite different, we perceive almost nothing about the noise except for its presence; we just recognize the pattern of the lines as the same and notice that there are some noise. Thus, to model the perception, we need a way to recognize the part of an infinite signal that represents finite but useful information. This is why we are especially interested in inherently finite structures whose literal manifestations are infinite.

The human visual system seems to have "the ability to impose organization on sensory data—to discover regularity, coherence, continuity, etc., on many levels," which is "apart from both the perception of tri-dimensionality and from the recognition of familiar objects[15]." We agree that such structure and organization that appears at every level is the key to modeling vision and perception in general. One purpose of this work is to provide a language to express the perceptual organization that enables us to implement the ability to impose it on the data.
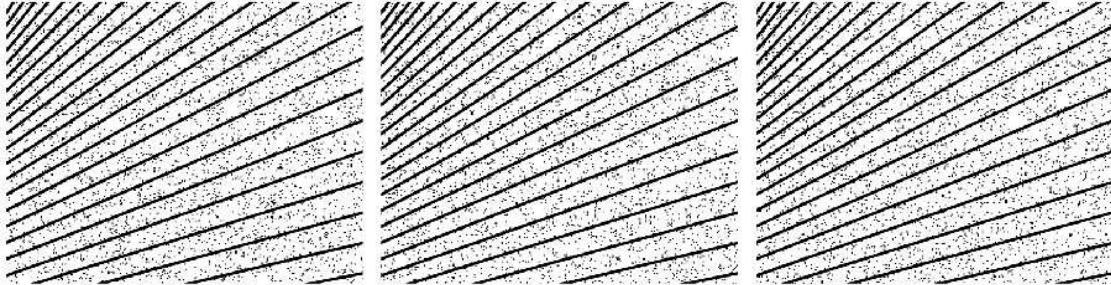
Figure 2: Visual patterns with noise.

1.3 Desiderata

What we seek is a description, or representation, of general objects with the following properties:

I. General: It can represent a general class of objects and all objects in the class, including a part of the descriptions themselves, allowing hierarchical description.

II. Uniform: It represents the objects in a uniform way by simple rules.

III. Reflect complexity: The intuitive complexity, or the amount of information in the object, corresponds to the complexity of the description. In particular, intuitively finite object has a finite description.

IV. Grounded: There is a set of rules that applies to the whole class of objects, not depending on the instance of the object, dictating how the described objects are related to the raw signal.

In the case of strings, the literal representation satisfies I and II: representing a string as a string is obviously general enough to represent any string, and the representation is uniform for any string. Fixing a universal Turing machine U, one can consider a program p for U as a description of a string x if p causes the machine to halt after writing out the string x on the tape. Intuitively simple string would have a shorter program. Also, describing strings by other strings automatically satisfy the describability of descriptions and the groundedness. Thus this description would satisfy all of the desiderata.

In the case of images, we can think of a function on a rectangle in $\mathbb{R}^2$ as the literal representation, satisfying the desiderata I and II. But we do not know of a representation that satisfies all of the desiderata. Perhaps the closest is the page description languages like PostScript, possibly modified to allow infinite precision. However, it has too many primitives to be convenient for mathematical treatment. Also, the uniformity and simplicity of

the rules of description is important not only for the sake of mathematical convenience, but also because we aim eventually to develop a way of automatically extract such description from the literal description, or the signal. More crucially, PostScript is not general enough: its class of objects is limited to two dimensional pages. The applicability to more general objects than just images is important because we would like to find a description that reflects the structure within more abstract data than a two-dimensional page, especially the description itself. For instance, if we have a way to describe circles by center points and radii, we would have a three-dimensional space of circles. We would like to use the same uniform description to describe a group of the 3D points corresponding to the circles. Thus, allowing "describing the description" is crucial in order to allow efficient description of repetitive, hierarchical, and more general structures.

The groundedness requirement is needed to treat general structures in a uniform way. When we say that some data represents some object, we implicitly assume a set of rules for data interpretation and manipulation. It is this set of rules that gives the structure to the object. It is like a machine with knobs and buttons to control it: knowing their settings may be enough to determine the state of the machine; but to describe the effect and interaction of the machine with the environment and other machines, we need more than the internal parameters. If the rules are *ad hoc*, varying from one instance of representation to another, it would be impossible to formulate the notion of general structures and describe the manipulation of and interaction between such structures.

All the desiderata are related to each other. In particular, we emphasize the following: it is not enough that the simple objects correspond to less data (III); the correspondence must be obtainable from the representation (IV) in a uniform (II) way that applies to the whole general class (I) of objects. For instance, we can say a pair of a point and a real number represents a circle by regarding them as the center and the radius. Or we can say that the pair represents a line by regarding it as a point the line goes through and its angular direction. But for the representation to cover both cases, one must also include in the representation some data specifying which case it is for each object. Such data quickly adds up when one wants to represent various shapes; so when we say that general shapes are described in a representation, it has not only to cover all the shapes but also include the necessary data in a way any shape represented can be converted into a common, literal representation.

1.4 Related Work

The General Pattern Theory[6, 7, 8] is an effort to provide an algebraic framework for describing patterns as structures. It defines a vocabulary which is manipulated to cast the concept of pattern in a precise algebraic language. While it has detailed algebraic and statistical theories with many examples, we only discuss here the part that deals with the representation of patterns. The representation is based on graphs. A graph is fixed; each of its node can be assigned one of *generators*, whose set is predetermined; a restriction as to

which combination of generators can be assigned to the nodes is defined as a set of pairwise restrictions corresponding to the edges of the graph. There are numerous examples in the literature showing that this representation can be used to represent many classes of objects.

The Syntactic Pattern Recognition[4] also represents patterns in a way that explicitly handles the interrelationships between the parts that make up the whole of the object, and use the explicit structure in patterns to recognize them. The representation is either by a formal language or a graph.

Neither of the representations used in the two formalisms is satisfactory for us. The crucial problem is that they are not grounded in the sense above. They are not uniform from a class of objects to another; thus, although we can talk about the information in objects in each class, there is no way to compare them across the classes. They are general in the sense one can adopt them to many different classes of objects, but they are not general enough to represent all of the classes in the same uniform way. They cannot be used, for instance, to define what patterns are, because there is no prescribed way for the representation to be connected to general enough class of objects. There is no formal way to give a raw data and ask what pattern it might form.

Besides the Kolmogorov complexity we already mentioned, there are many notions of the complexity of objects. Most of them are concerned about the complexity of objects that do not include what we deal with in this paper. Also, note that what we define in this paper is a measure of *information* like Kolmogorov complexity and Shannon information[12], rather than a measure of *complexity* such as the computational complexity. We refer the reader to the appendix of [3] for an overview of formulations of complexity with an extensive bibliography.

1.5   Overview of the Representation

In this paper, we introduce a representation that fulfills all the desiderata above. Here, we give an overview of its definition and some of its properties.

We assume that the objects are given *a priori* as subsets of some sets. Those that can be thought of in this way forms a very general class that seems to include most, if not all, objects we might deal with. For example, we can think of a binary string $s = s_0 s_1 \cdots s_n$ as a subset $\{(i, s_i) \mid i = 0, \cdots, n\}$ of $\mathbb{N} \times \{0, 1\}$; an image can be thought of as a subset of the product of the image plane and the color space, i.e., the graph of the image function on the image plane. A physical object like a bicycle or an automobile, at one level of abstraction, can be thought of as a subset of $E^3 \times M$, where E^3 is the 3D Euclidean space and M the set of materials, e.g., glass, iron, rubber, etc.; the subset consists of $(x, m(x))$, with $m(x)$ the material that occupies the point x in E^3. We call this representation of objects as subsets the *ground representation*; it serves as a signal-level, literal representation. It is simple to represent something in this way; we can then ask the amount of information therein.

The ground representation is an abstraction of the kind of data representation that we call the *dense* representation, which includes strings, bitmaps, and other raw data. It cor-

responds to representing a string as itself. One property of dense representation is that the presence of regularities does not affect it. For instance, any image can be represented as a bitmap in exactly the same manner, whether it is a regular image or a white noise. Another type of representation, which we call the *sparse* representation, utilizes regularities in the object to describe it. In the image example, if it is an image of geometric objects, it should take very little data to describe it, at least in principle; a circle, for instance, can be represented just by specifying its center point and radius. The same kind of description cannot be used to describe a white noise.

An important feature of the representation proposed in this paper is that it can interpolate between the dense and sparse representation, so that it can take advantage of regularity in the data while also being capable of representing any, even random, data. This is similar to using an input to a Turing machine to represent strings.

As the vocabulary to describe the regularities in such objects, we use the maps that characterize the space in which the objects are included as subsets. Maps characterize the structure of spaces in the following sense. Any two sets with the same cardinality are the same sets in absence of other characteristics. For instance, $\mathbb{R}$ is the same set in this sense as $\mathbb{R}^2$, i.e., there exists a one-to-one map between them, if we disregard the structures such as the topology, the vector space structure, the metric structure, the order, and the algebraic structure. These structures can be characterized by maps. For instance, the metric structure is defined by the distance map that gives the distance between two elements of the set; the order is given by a predicate on a pair of elements that returns *true* if the first element is less than the second. The two sets are different when we consider the structures because the one-to-one map does not commute with the maps that define the structures.

Using such *structure maps* to describe regularity, objects with regularities can be represented through sparse parameters in the proposed representation. For a given object, there can be many different ways of representing it, just as there can be any number of inputs to a Turing machine producing the same string. Importantly, there is a prescribed way to connect the description to the ground representation; thus, the representation is grounded. While taking structures into account so that regular objects can be represented as such, it automatically provides an interpretation of each represented object into the signal level. In other words, the relationship between the parameters and the data is part of the representation. Thus, we can give our data in the ground representation and then ask what sparser, more structured representation is possible.

Let us be slightly more concrete. In the proposed representation, we take a number of sets and maps between them (that are to be composed by the structure maps), which we call a *diagram*. Then we call an assignment of a subset to each set in the diagram its *cross section*. A cross section must satisfy a certain constraints, because of which we can uniquely determine all the subsets by specifying only a partial cross section, which assigns subsets to only some of the sets in the diagram. If one of the subsets coincides with the ground representation of the object in question, we say that it is represented by the diagram and the partial cross section.

9

The maps define the structures we take into consideration, which determine the regularities, which in turn allow more concise description of the object than the literal one. Since the representation by diagrams and cross sections is explicitly in terms of the maps in the diagram, it is apparent from the diagram exactly what structure is taken into account.

Some of the properties of the representation are as follows. It is:

i) Sparse: Unlike dense representations such as bitmap, it is capable of representing objects by a combination of their essential structure and instance-specific parameters. The diagram expresses the essential structure while the partial cross section represents the parameters. Implementation-dependent approximation of representation only affects the parameters and thus can be separated from the structure. The sparseness also makes it flexible and easy to manipulate. By modifying the parameters, different instances of the same structure can be easily represented. Also, comparison of two patterns having the same structure is naturally defined.

ii) Modular: Parts of the representation can be understood as the modules to construct larger and more complex ones. Complex combinations can be obtained hierarchically and recursively as well as by simple union and intersection.

iii) Hierarchical: Because it can be applied to any data, it can also be applied to the parameter space parametrizing some other structure, leading to a hierarchical representation.

iv) Recursive: It can represent a recursively defined structure, making it particularly powerful in, for instance, representing repeated patterns. The "repeat" can be in various spaces that can manifest in the final pattern in non-obvious ways.

Finally, diagrams and their cross sections can represent maps between powersets. In fact, the representation of subsets can be considered a special case where the map sends a trivial set to the subset. Any computation, in particular, can also be represented.

1.6 Measure of Structural Information

Using the representation, we introduce a measure of information. Roughly speaking, it is defined as the size of the smallest diagram representing the object, where the diagram can contain only those maps that can be composed by a set of given structure maps, including constant maps.

Thus, the measure is relative to the structure and constants expressed explicitly in the form of maps. The explicit incorporation of the structure of the object space is the key to avoiding the trap of arbitrariness. The patterns such as shown in Figure 1 all have finite information according to the measure. The measure is relative to the constants because of the aforementioned need to separate the information in the structure from that in infinite objects such as real numbers.

Because of the reasons laid out in 1.1, we do not follow the recipe of interfacing Kolmogorov complexity by encoding objects by strings; strings are given no special status in

10

this theory. Instead, we define it directly in terms of the structure of the spaces in which the objects reside. As such, the new measure does not depend on Kolmogorov complexity. Since the class of applicable objects includes that of strings, however, it raises a question of their relationship. It turns out that the new measure is equivalent to Kolmogorov complexity in the case where strings are characterized by the structure of natural numbers given by the constant 0 and the successor function. Note that this is not obvious *a priori*: the definition of the representation and the measure does not even mention strings. Also, the new measure is defined relative to structure maps. It is equivalent to Kolmogorov complexity when it is defined relative to this particular set of structure maps; relative to other sets, it may not be. If the set includes the constant maps of all strings, for example, any string's information would be 1.

Thus, the new measure gives the larger class of objects a meaningful measure of information that generalizes Kolmogorov complexity.

The rest of the paper is organized as follows. In the next section, we define the notion of diagrams and their cross sections precisely, as well as what is meant by representing with them. We also list the notations used throughout this paper. In section 3, we illustrate some properties of the representation by geometric examples. In section 4, we give more examples, this time those representing computations. In section 5, we define the information measure of structure of general objects. In section 6, we prove that the measure generalizes Kolmogorov complexity. In section 7, we further discuss the difference between our approach and the string-centered one, before concluding.

2 Representation by Diagrams and Cross Sections

2.1 Definitions

We fix the notation for standard finite sets as $\mathbf{1} = \{0\}, \mathbf{2} = \{0, 1\}, \cdots, \boldsymbol{n} = \{0, 1, \cdots, n-1\}$, etc. The set $\mathbf{2}$ is also used as the set of Boolean values, 0 meaning *false* and 1 *true*. We mean by $f : X \to Y$ that f is a map from X to Y. We denote the set of all subsets of X (the power set of X) by $\mathbf{2}^X$. The map from $\mathbf{1}$ to X that maps $0 \in \mathbf{1}$ to $x \in X$ is denoted by the same letter $x : \mathbf{1} \to X$. We call it a *constant map*.

Definition 1. Let $\mathscr{S} = (S_i)_{i \in \mathscr{I}}$ be a family of sets indexed by a set $\mathscr{I}$. A *cross section s* of $\mathscr{S}$ is an assignment to each set S_i in $\mathscr{S}$ of its subset $s_i \subset S_i$.

In other words, a cross section of $\mathscr{S}$ is another family $(s_i)_{i \in \mathscr{I}}$ of sets indexed by $\mathscr{I}$ such that $s_i \subset S_i$ for all $i \in \mathscr{I}$. We used the index set to make clear that there can be multiple members of the family $\mathscr{S}$ that are identical as sets; however, we avoid the use of indices almost entirely in this paper. We use the set-theoretic notation with $\mathscr{S}$ such as $S \in \mathscr{S}$. The equality of two members of $\mathscr{S}$ means that their indices are the same; if the indices are different, we treat them as different, even if they are identical as sets. When we discuss a

set $S = S_i$ in $\mathscr{S}$ and a cross section s of $\mathscr{S}$, $s(S)$ denotes the subset s_i assigned to S_i by s. Thus, s assigns each $S \in \mathscr{S}$ its subset $s(S) \subset S$.

We denote the set of cross sections of $\mathscr{S}$ by $\Gamma(\mathscr{S})$. Let $\mathscr{T}$ be a subfamily of $\mathscr{S}$. A cross section of $\mathscr{T}$ is called a partial cross section of $\mathscr{S}$. For a cross section s of $\mathscr{S}$, the cross section of $\mathscr{T}$ that assigns $s(S)$ to S in $\mathscr{T}$ is called the restriction of s to $\mathscr{T}$, denoted by $s|_{\mathscr{T}}$. For a cross section t of $\mathscr{T}$, we denote the set $\{s \in \Gamma(\mathscr{S}) \mid s|_{\mathscr{T}} = t\}$ of cross sections of $\mathscr{S}$ that restrict to t by $\Gamma(\mathscr{S} \mid t)$.

Definition 2. A *diagram* is a triple $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ of a family $\mathscr{S} = (S_i)_{i \in \mathscr{I}}$ of sets, its subfamily $\mathscr{S}'$, and a family $\mathscr{M} = (\varphi_j)_{j \in \mathscr{J}}$ of maps of the form $\varphi_j : 2^S \to 2^T$, with $S, T \in \mathscr{S}$, where $\mathscr{I}$, $\mathscr{J}$ are index sets.

A diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ such that both $\mathscr{S}$ and $\mathscr{M}$ are finite is called a finite diagram. Let $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ be a diagram. There are maps $\mathrm{dm} : \mathscr{M} \to \mathscr{S}$ and $\mathrm{cdm} : \mathscr{M} \to \mathscr{S}$ such that $\varphi : 2^{\mathrm{dm}(\varphi)} \to 2^{\mathrm{cdm}(\varphi)}$ for $\varphi \in \mathscr{M}$. Also, define the maps $\mathrm{out} = \mathrm{dm}^{-1} : \mathscr{S} \to 2^{\mathscr{M}}$ and $\mathrm{in} = \mathrm{cdm}^{-1} : \mathscr{S} \to 2^{\mathscr{M}}$ so that $\mathrm{out}(S) = \{\varphi \in \mathscr{M} \mid \mathrm{dm}(\varphi) = S\}$ and $\mathrm{in}(S) = \{\varphi \in \mathscr{M} \mid \mathrm{cdm}(\varphi) = S\}$ for $S \in \mathscr{S}$.

Definition 3. The *cross section of a diagram* $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ is a cross section s of $\mathscr{S}$ such that, for any $S \in \mathscr{S}$ with $\mathrm{in}(S) \neq \emptyset$, the following holds:

$$s(S) = \bigcap_{\varphi \in \mathrm{in}(S)} \varphi(s(\mathrm{dm}(\varphi))) \text{ if } S \in \mathscr{S} \setminus \mathscr{S}', \tag{1}$$

$$s(S) = \bigcup_{\varphi \in \mathrm{in}(S)} \varphi(s(\mathrm{dm}(\varphi))) \text{ if } S \in \mathscr{S}'. \tag{2}$$

In diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$, the subfamily $\mathscr{S}'$ of $\mathscr{S}$ specifies the sets for which a cross section should satisfy (2) instead of (1); this means that the cross section on that set should be the union, rather than the intersection, of the images by the incoming maps. We denote the set of cross sections of diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ by $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M})$. We also define $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t) = \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M}) \cap \Gamma(\mathscr{S} \mid t)$ for $\mathscr{T} \subset \mathscr{S}$ and $t \in \Gamma(\mathscr{T})$; i.e., $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$ is the set of cross sections of diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ that restrict to the cross section t of subfamily $\mathscr{T}$ of $\mathscr{S}$.

To illustrate the definitions by example, suppose $\mathscr{S} = \{\mathbf{1}, X, Y, Z, W\}$, $\mathscr{S}' = \{W\}$, $w \in W$, and $\mathscr{M} = \{w, \varphi, \psi, \eta, \delta, \kappa\}$ with

$$w : 2^{\mathbf{1}} \to 2^W, \qquad \varphi : 2^Y \to 2^X, \qquad \psi : 2^Z \to 2^Y,$$
$$\eta : 2^W \to 2^Y, \qquad \delta : 2^W \to 2^X, \qquad \kappa : 2^X \to 2^W,$$

where w is an element of W; the same letter denotes a constant map. We denote the diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ as follows:

$$\tag{3}$$

12
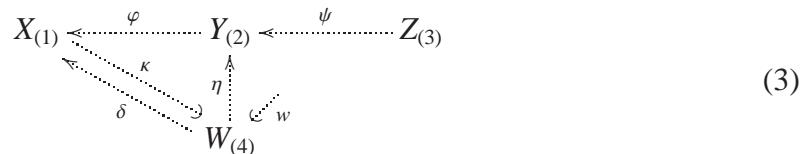
For instance, φ maps each subset of Y to a subset of X. We omit the set $\mathbf{1}$ from the diagram: a constant map is shown as an incoming arrow, without the domain $\mathbf{1}$. Note also that the arrows have dotted shafts, which signifies that the map is between power sets. The parenthesized subscript numbers are for reference: as more than one sets in the family can be identical as sets, we use these to refer to them. We always use S_i to mean the set with the subscript (i) in the diagram under discussion. Thus, if we are discussing the one in equation (3), S_1 means the set X and S_2 means Y, etc. Also, there are two kinds of arrowheads: the ordinary arrows and round ones. An arrow has the round arrowhead if and only if it is coming into a set in $\mathscr{S}'$, in this case $S_4 = W$. For a cross section s of this diagram, we have, e.g.,

$$\mathrm{in}(X) = \{\varphi, \delta\}, \quad \mathrm{dm}(\varphi) = Y, \quad s(X) = \varphi(s(Y)) \cap \delta(s(W)), \quad s(W) = \kappa(s(X)) \cup \{w\}.$$

A diagram and its partial cross section represents an object in the following sense.

Definition 4. Let $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ be a diagram, X a set in $\mathscr{S}$, $\mathscr{T}$ a subfamily of $\mathscr{S}$, and t a cross section of $\mathscr{T}$. Suppose an object is represented in the ground representation as a subset A of X. The object is said to be represented by $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, X)$ if $s(X) = A$ for every cross section s in $(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$.

The ground representation is a special case of this as a *trivial representation*; just take $\mathscr{S} = \mathscr{S}' = \mathscr{T} = \{X\}$, $\mathscr{M} = \emptyset$, and $t(X) = A$. Thus the representation is general enough to include all dense representation. The aim, however, is to enable more efficient representation that captures the structure.

Here, we also define the concepts of minimality and limit for later use.

Definition 5. Let $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ be a diagram, $X \in \mathscr{S}$, and $G \subset \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M})$. A cross section $s \in G$ such that no other $t \in G$ gives $t(X) \subsetneq s(X)$ is said to be *minimal* on X in G. We denote the set of such cross sections by $\min_X G$.

Note that $\min_Y \min_X G \supset \min_X G \cap \min_Y G$ for $X, Y \in \mathscr{S}$: if $s \in \min_X G \cap \min_Y G$, then $s(Y) \subset t(Y)$ for any $t \in \min_X G$ since $t \in G$ and $s \in \min_Y G$; thus $s \in \min_Y \min_X G$. Since it is also the case that $\min_Y \min_X G \subset \min_X G$, by symmetry it follows that $\min_Y \min_X G \cap \min_X \min_Y G = \min_X G \cap \min_Y G$.

Definition 6. Let $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ be a diagram, $X \in \mathscr{S}$, $\mathscr{T} \subset \mathscr{S}$, and $t \in \Gamma(\mathscr{T})$. Furthermore, let $X_1, \cdots, X_n$ be a finite number of sets in $\mathscr{S}$. A subset A of X is said to be *represented by the data $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, X, (X_1, \cdots, X_n))$ as a limit* if $s(X) = A$ for any cross section s in $\min_{X_n} \min_{X_{n-1}} \cdots \min_{X_1} \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$.

2.2 Notations

Here we list some more notations used in this paper.

13

i) For any set X, id : $X \to X$ denotes the identity map on X and $\omega : X \to \mathbf{1}$ the unique map from X to $\mathbf{1} = \{0\}$. The complement map cmpl : $2^X \to 2^X$ is defined for $A \subset X$ by:

$$\text{cmpl}(A) = {}^c A = X \setminus A. \tag{4}$$

ii) The product map $f_1 \times \cdots \times f_n : X \to Y_1 \times \cdots \times Y_n$ of maps $f_i : X \to Y_i$ ($i = 1, \cdots, n$) is defined by $(f_1 \times \cdots \times f_n)(x) = (f_1(x), \cdots, f_n(x))$. Given a map $f : X \to Y$ and a constant map $z : \mathbf{1} \to Z$, one can construct a product map:

$$f \times (z \circ \omega) : X \to Y \times Z$$

of f and $z \circ \omega : X \to Z$. By abuse of notation, we denote this map by $f \times z : X \to Y \times Z$. Similarly, we mix maps of the form $X \to Y$ and $\mathbf{1} \to Z$ freely in making a product map.

iii) For a Cartesian product $X_1 \times X_2 \times \cdots \times X_n$, the map

$$\pi_i : X_1 \times X_2 \times \cdots \times X_n \to X_i$$

is the projection to the i'th component. We use a shorthand π_{ij} for the product map $\pi_i \times \pi_j : X_1 \times \cdots \times X_n \to X_i \times X_j$, π_{ijk} for $\pi_i \times \pi_j \times \pi_k$, and so on.

iv) For a disjoint union $X_1 + X_2 + \cdots + X_n$, the map

$$\iota_i : X_i \to X_1 + X_2 + \cdots + X_n$$

is the injection from the i'th component.

v) The map union $f + g : X + Y \to Z$ of maps $f : X \to Z$ and $g : Y \to Z$ is defined by $(f + g)(x) = f(x)$ if $x \in X$ and $(f + g)(x) = g(x)$ if $x \in Y$.

vi) For a map $f : X \to Y$, we denote by the same letter the map $f : 2^X \to 2^Y$ between the power sets defined by $f(A) = \{f(x) \mid x \in A\} \subset Y$ for $A \subset X$.

vii) For a map $f : X \to Y$, the map $f^{-1} : 2^Y \to 2^X$ is defined by $f^{-1}(A) = \{x \in X \mid f(x) \in A\} \subset X$ for $A \subset Y$. By a slight abuse of notation, by $f^{-1}(y)$ for $y \in Y$ we mean $f^{-1}(\{y\})$.

viii) For a map $f : X \to X$, f^0 denotes id_X. For a positive integer n, f^n denotes the map $f^n : X \to X$ defined as applying f for n times as well as the map $f^n : 2^X \to 2^X$ defined as in vi). When n is a negative integer, f^n denotes the map $f^n : 2^X \to 2^X$ defined as applying $f^{-1} : 2^X \to 2^X$ for $-n$ times.

3 Geometric Patterns

Using diagrams and cross sections, we can represent geometric objects in a uniform and compact way. In this section, we introduce the representation and discuss its properties using examples.

14

3.1 Examples

As the simplest example, we consider a circle in the Euclidean plane X. Let us denote the vector space of translations in X by V. Also, denote the map that sends $(x, y) \in X \times X$ to $x - y$ by sub $: X \times X \to V$, and the map that gives the length of a vector by len $: V \to \mathbb{R}$.

Consider the following diagram:

$$
\begin{array}{ccc}
\mathbb{R}_{(1)} & \xdashrightarrow{\text{len}^{-1}} & V_{(2)} \\
& & \Big\downarrow{\text{sub}^{-1}} \\
X_{(3)} & \xdashrightarrow{\pi_2^{-1}} X \times X_{(4)} & \xdashrightarrow{\pi_1} X_{(5)}
\end{array}
\tag{5}
$$

This denotes a diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ with

$$
\mathscr{S} = (S_1, \cdots, S_5), \quad \mathscr{S}' = \emptyset, \quad S_1 = \mathbb{R}, \quad S_2 = V, \quad S_3 = S_5 = X, \quad S_4 = X \times X
$$

and

$$
\mathscr{M} = (\text{len}^{-1} : 2^{S_1} \to 2^{S_2}, \text{sub}^{-1} : 2^{S_2} \to 2^{S_4}, \pi_2^{-1} : 2^{S_3} \to 2^{S_4}, \pi_1 : 2^{S_4} \to 2^{S_5}).
$$

Note that, while the inverse maps are indicated by $^{-1}$, the power map in the forward direction is to be surmised from the convention that the map is between powersets.

Suppose that $\mathscr{T} = \{S_1, S_3\}$ and that its cross section t is defined by

$$
t(S_1) = \{r\}, \quad t(S_3) = \{p\},
$$

where r is a positive real number and p is a point in the Euclidean plane X. Let s be a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$. Then, by (1), we have

$$
\begin{aligned}
s(S_2) &= \text{len}^{-1}(s(S_1)) = \text{len}^{-1}(t(S_1)) = \text{len}^{-1}(\{r\}) = \{v \in V \mid \text{len}(v) = r\}, \\
s(S_4) &= \text{sub}^{-1}(s(S_2)) \cap \pi_2^{-1}(s(S_3)) \\
&= \{(x, y) \in X \times X \mid x - y \in s(S_2), y \in s(S_3)\}, \\
s(S_5) &= \{\pi_1((x, y)) \in X \mid (x, y) \in s(S_4)\} \\
&= \{x \in X \mid x - y \in s(S_2), y \in s(S_3)\} \\
&= \{x \in X \mid \text{len}(x - p) = r\}.
\end{aligned}
$$

Thus the cross section s is completely determined and $s(S_5)$ is the set of the points on the circle centered at p with radius r. In this way, $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_5)$ represents the circle.

If $t(S_3) = \{p, q\}$, it represents two circles with the same radius r centered at p and q. Thus, we can think of S_3 as the space of centers of the circles. If $t(S_1) = \{r, t\}$ instead, it would represent two concentric circles with radii r and t. If we modify the diagram to

$$
\mathbb{R} \times X_{(1)} \xdashrightarrow{((\text{len} \circ \pi_1) \times \pi_2)^{-1}} V \times X_{(2)} \xdashrightarrow{(\text{sub} \times \pi_2)^{-1}} X \times X_{(3)} \xdashrightarrow{\pi_1} X_{(4)}
$$

and let $\mathscr{T} = \{S_1\}$ and define $t \in \Gamma(\mathscr{T})$ by $t(S_1) = \{(r_1, p_1), (r_2, p_2), \cdots\}$, then we have

$$s(S_2) = \{(v, x) \in V \times X \mid (\text{len}(v), x) \in t(S_1)\},$$
$$s(S_3) = \{(x, y) \in X \times X \mid (x - y, y) \in s(S_2)\},$$
$$s(S_4) = \{x \in X \mid {}^{\exists}y \in X, (\text{len}(x - y), y) \in t(S_1)\},$$

and we have as $s(S_4)$ the circles specified by the radius-center pairs in $t(S_1)$.

For another example, a line in X can be represented using the following diagram:

$$
\begin{array}{ccccc}
V_{(1)} & \xrightarrow{\ \pi_1^{-1}\ } & V \times \mathbb{R}_{(2)} & & \\
& & \Big\downarrow{\scriptstyle \text{sub}^{-1}\,\circ\,\text{mult}} & & \\
X_{(3)} & \xrightarrow{\ \pi_2^{-1}\ } & X \times X_{(4)} & \xrightarrow{\ \pi_1\ } & X_{(5)}
\end{array}
\tag{6}
$$

Here, mult $: V \times \mathbb{R} \to V$ is the scalar multiplication. Other maps are as above. Suppose that $\mathscr{T} = \{S_1, S_3\}$ and that its cross section t is defined by

$$t(S_1) = \{v\}, \quad t(S_3) = \{p\},$$

where p is a point in the Euclidean plane X and v is a vector in V. Let s be a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$. Then, from (1) we have

$$
\begin{aligned}
s(S_2) &= \pi_1^{-1}(\{v\}) = \{(v, c) \in V \times \mathbb{R} \mid c \in \mathbb{R}\}, \\
s(S_4) &= \text{sub}^{-1}(\text{mult}(s(S_2))) \cap \pi_2^{-1}(\{p\}) \\
&= \text{sub}^{-1}(\{cv \in V \mid c \in \mathbb{R}\}) \cap \pi_2^{-1}(\{p\}) \\
&= \{(x, p) \in X \times X \mid {}^{\exists}c \in \mathbb{R}, x - p = cv\}, \\
s(S_5) &= \{x \in X \mid {}^{\exists}c \in \mathbb{R}, x - p = cv\} \\
&= \{p + cv \mid c \in \mathbb{R}\}.
\end{aligned}
$$

Thus, the cross section s is completely determined and $s(S_5)$ consists of the points on the line that goes through p and has the direction parallel to v.

3.2 Union

As mentioned in 2.1,

$$S_1 \xdashrightarrow{\ \phi\ } S_2 \xdashleftarrow{\ \psi\ } S_3$$

denotes the case when (1) in Definition 3 is required, i.e., $S_2 \in \mathscr{S} \setminus \mathscr{S}'$. Any cross section s of the diagram satisfies $s(S_2) = \phi(s(S_1)) \cap \psi(s(S_3))$. To denote the other case, we use

$$S_1 \xdashrightarrow{\ \phi\ } S_2 \xdashleftarrow{\ \psi\ } S_3 \tag{7}$$

16

to indicate that $S_2 \in \mathscr{S}'$ and $s(S_2) = \phi(s(S_1)) \cup \psi(s(S_3))$. Thus, for any set in $\mathscr{S}'$, incoming maps are depicted with the round arrow.

In the examples, we may use two kinds of incoming arrows as:

$$S_1 \xrightarrow{\varphi} S_2 \xleftarrow{\psi} S_3$$

$$\text{with } \theta \text{ from } S_4 \text{ and } \eta \text{ from } S_5$$
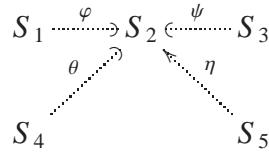
It means $s(S_2) = (\varphi(s(S_1)) \cup \psi(s(S_3)) \cup \theta(s(S_4))) \cap \eta(s(S_5))$, i.e., we take the unions first, and then the intersection. This is simply an abbreviation of

$$S_1 \xrightarrow{\varphi} S_6 \xleftarrow{\psi} S_3$$
$$\theta \quad \downarrow \text{id}$$
$$S_4 \qquad S_2 \xleftarrow{\eta} S_5$$
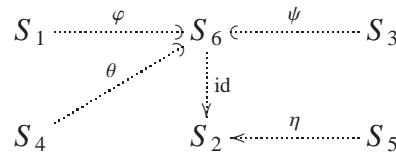
If we allow the complement map $\mathrm{cmpl} : 2^X \to 2^X$ in diagrams, we need only one of the conditions in Definition 3, because we can make unions from intersections or vice versa. Using cmpl,

$$S_1 \xrightarrow{\mathrm{cmpl} \circ \phi} S_4 \xleftarrow{\mathrm{cmpl} \circ \psi} S_3$$
$$\downarrow \mathrm{cmpl} \tag{8}$$
$$S_2$$
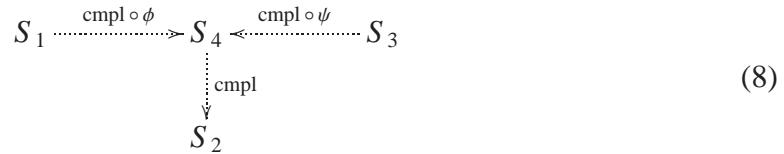
$$s(S_4) = {}^c\phi(s(S_1)) \cap {}^c\psi(s(S_3))$$
$$s(S_2) = {}^c s(S_4) = \phi(s(S_1)) \cup \psi(s(S_3)).$$

Thus, (8) is equivalent to (7).

3.3 Representing maps

A diagram with partial cross section can represent a map in the following sense:

Definition 7. A map $\varphi : 2^X \to 2^Y$ is said to be represented by $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, X, Y)$ if $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ is a diagram, $\mathscr{T} \subset \mathscr{S}$, $t \in \Gamma(\mathscr{T})$, $X, Y \in \mathscr{S}$, and every cross section s in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$ satisfies $s(Y) = \varphi(s(X))$.

As an example, let us represent the map $\mathrm{cl} : 2^X \to 2^X$ that maps a subset A of a Euclidean space X to the topological closure $\bar{A}$ of A in X. Consider the diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:

$$X_{(1)} \xrightarrow{\pi_1^{-1}} X \times X_{(2)} \xrightarrow{\mathrm{sub} \times \pi_2} V \times X_{(3)} \xrightarrow{(\mathrm{len} \circ \pi_1) \times \pi_2} \mathbb{R} \times X_{(4)} \xrightarrow{\mathrm{Inf}} \mathbb{R} \times X_{(5)}$$
$$\pi_1^{-1} \nearrow \qquad \downarrow \pi_2$$
$$\mathbb{R}_{(6)} \qquad\qquad X_{(7)}$$

17
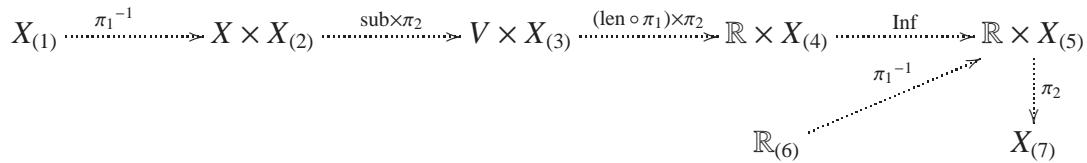
Here, the map Inf maps $B \subset \mathbb{R} \times X$ to $\{(a, x) \in \mathbb{R} \times X \mid x \in \pi_2(B), a = \inf \pi_1(B \cap \pi_2^{-1}(x))\}$; thus, for each $x \in X$ that appears in B, there is an element (a, x) in $\mathrm{Inf}(B)$, where a is the infimum of the set of real numbers b that appear as (b, x) in B. Now, if $\mathscr{T} = \{S_6\}, t(S_6) = \{0\}, s \in \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid \mathscr{T})$, and $s(S_1) = A$, we have

$$s(S_2) = \{(x, y) \mid x \in A, y \in X\}$$
$$s(S_4) = \{(d, y) \mid y \in X, \exists x \in A, \mathrm{len}(x - y) = d\}$$
$$s(S_7) = \{y \in X \mid \inf_{x \in A} \mathrm{len}(x - y) = 0\} = \bar{A}.$$

Thus $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_1, S_7)$ represents cl.

The infinimum map Inf in turn can be represented by

$$
\begin{array}{ccccccc}
\mathbb{R} \times X_{(1)} & \xrightarrow{\pi_{13}^{-1}} & \mathbb{R} \times \mathbb{R} \times X_{(2)} & \xrightarrow{\pi_2 \times (\mathrm{lt} \circ \pi_{12}) \times \pi_3} & \mathbb{R} \times \mathbf{2} \times X_{(3)} & \xrightarrow{\mathrm{cmpl} \circ \pi_{13}} & \mathbb{R} \times X_{(4)} \\
 & & & & \uparrow{\scriptstyle \pi_2^{-1}} & & \downarrow{\scriptstyle \mathrm{Max}} \\
 & & & & \mathbf{2}_{(5)} & & \mathbb{R} \times X_{(6)}
\end{array}
$$

with $\mathscr{T} = \{S_5\}, t(S_5) = \{1\}$. The map $\mathrm{lt} : \mathbb{R} \times \mathbb{R} \to \mathbf{2}$ maps (a, b) to 1 if $a < b$ and 0 otherwise, while the map Max maps $B \subset \mathbb{R} \times X$ to $\{(a, x) \in \mathbb{R} \times X \mid x \in \pi_2(B), a = \max \pi_1(B \cap \pi_2^{-1}(x))\}$. Then if $s \in \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid \mathscr{T})$ and $s(S_1) = B$ we have

$$s(S_2) = \{(a, b, x) \mid (a, x) \in B, b \in \mathbb{R}\},$$
$$s(S_3) = \{(b, 1, x) \mid \exists (a, x) \in B, a < b\},$$
$$s(S_4) = \{(b, x) \mid \nexists (a, x) \in B, a < b\},$$
$$s(S_6) = \{(c, x) \mid x \in \pi_2(B), c = \max\{b \in \mathbb{R} \mid \nexists (a, x) \in B, a < b\}\}$$
$$= \{(c, x) \mid x \in \pi_2(B), c = \inf(B \cap \pi_2^{-1}(x))\}.$$

Thus $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_1, S_6)$ represents Inf.

Finally, the maximum map Max can be represented by

$$
\begin{array}{ccc}
\mathbb{R} \times X_{(1)} & \xrightarrow{\mathrm{id}} & \mathbb{R} \times X_{(2)} \\
{\scriptstyle \pi_{13}^{-1}} \downarrow \downarrow {\scriptstyle \pi_{23}^{-1}} & & \uparrow{\scriptstyle \mathrm{cmpl} \circ \pi_{13}} \\
\mathbb{R} \times \mathbb{R} \times X_{(3)} & \xrightarrow{\pi_1 \times (\mathrm{lt} \circ \pi_{12}) \times \pi_3} \mathbb{R} \times \mathbf{2} \times X_{(4)} \xleftarrow{\pi_2^{-1}} & \mathbf{2}_{(5)}
\end{array}
$$

with $\mathscr{T} = \{S_5\}, t(S_5) = \{1\}$. Then if $s \in \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid \mathscr{T})$ and $s(S_1) = B$ we have

$$s(S_3) = \{(a, b, x) \mid (a, x), (b, x) \in B\},$$
$$s(S_4) = \{(a, 1, x) \mid (a, x) \in B, \exists (b, x) \in B, a < b\},$$
$$s(S_2) = \{(a, x) \in B \mid \nexists (b, x) \in B, a < b\}.$$

Thus $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_1, S_2)$ represents Max.

3.4 Recursive Definition

Consider the following diagram:

$$
\begin{array}{ccc}
V_{(1)} & \xrightarrow{\ \pi_2^{-1}\ } & X \times V_{(2)} \\
& & \Big\uparrow{\scriptstyle \pi_1^{-1}} \ \Big\downarrow{\scriptstyle \text{add}} \\
X_{(3)} & \xrightarrow{\ \text{id}\ } & X_{(4)}
\end{array}
\tag{9}
$$

Here, $\text{add} : X \times V \to X$ is the parallel translation $(x, w) \mapsto x + w$ in the Euclidean space X. Suppose that $\mathscr{T} = \{S_1, S_3\}$ and that its cross section t is defined by

$$
t(S_1) = \{v\}, \quad t(S_3) = \{p\},
\tag{10}
$$

where p is a point in the Euclidean plane X and v is a vector in V. Let s be a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \,|\, t)$. Then, from (1) we have

$$
\begin{aligned}
s(S_2) &= \{(x, w) \mid x \in s(S_4), w \in s(S_1)\}, \\
s(S_4) &= \{p\} \cup \{x + w \mid (x, w) \in s(S_2)\} \\
&= \{p\} \cup \{x + w \mid x \in s(S_4), w \in s(S_1)\}
\end{aligned}
\tag{11}
$$

From (11), clearly $s(S_4) \supset D = \{p, p + v, p + 2v, p + 3v, \cdots\}$, i.e., $s(S_4)$ contains equally spaced points beginning at p and separated by v. However, this does not uniquely determine the cross section: for instance, we can define $s(S_4) = X$; or indeed any set that is the union of D and a set invariant under the translation by v.

To make it unique, we can take $\min_{S_4} \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \,|\, t)$. Then it only contains the cross section with $s(S_4) = D$.

Or we can use the following proposition. Let $\mathbb{N} = \{0, 1, 2, 3, \cdots\}$ denote the set of natural numbers.

Proposition 1. *Suppose that a set S has a "grading" function $g : S \to \mathbb{N}$ and let S_n denote $g^{-1}(n)$ for $n \in \mathbb{N}$. Consider a map $\eta : 2^S \to 2^S$ that satisfies, for $i \in \mathbb{N}$,*

$$
\eta(S_i) \subset S_{i+1}, \qquad \eta(S) = \bigcup_{n=0}^{\infty} \eta(S_n).
$$

If S can be written $S = S_0 \cup \eta(S)$, then

$$
S = \bigcup_{n=0}^{\infty} \eta^n(S_0).
$$

Proof. Since $S_n \neq S_m$ if $n \neq m$, for $x \in S_{n+1}$ with $n \in \mathbb{N}$, $x \notin S_0$ and $x \notin \eta(S_m) \subset S_{m+1}$ if $m \neq n$. Thus $x \in \eta(S_n)$ follows from

$$
S = S_0 \cup \eta(S) = S_0 \cup \bigcup_{n=0}^{\infty} \eta(S_n).
$$

Thus $S_{n+1} \subset \eta(S_n)$. Since $\eta(S_n) \subset S_{n+1}$, it follows $\eta(S_n) = S_{n+1}$. Therefore, $S_n = \eta(S_{n-1}) = \eta(\eta(S_{n-2})) = \cdots = \eta^n(S_0)$. The proposition follows from

$$S = \bigcup_{n=0}^{\infty} S_n.$$

$\square$

To use Proposition 1, we modify (9) as:

$$
\begin{array}{ccc}
V_{(1)} & \xrightarrow{\;\pi_2^{-1}\;} & X \times V \times \mathbb{N}_{(2)} \\
 & & \uparrow \varphi \\
\pi_{13}^{-1} & & \\
X \times \mathbb{N}_{(3)} & \xrightarrow{\;\text{id}\;} X \times \mathbb{N}_{(4)} \xrightarrow{\;\pi_1\;} & X_{(5)}
\end{array}
\tag{12}
$$

and define $\varphi : X \times V \times \mathbb{N} \to X \times \mathbb{N}$ by $\varphi((x, w, k)) = (x + w, k + 1)$ as well as modifying (10) to $t(S_1) = \{v\}$, $t(S_3) = \{(p, 0)\}$. Then (11) becomes

$$
\begin{aligned}
s(S_4) &= s(S_3) \cup \varphi(s(S_2)) \\
&= \{(p, 0)\} \cup \{(x + w, k + 1) \mid (x, k) \in s(S_4), w \in s(S_1)\}.
\end{aligned}
\tag{13}
$$

We define $g : s(S_4) \to \mathbb{N}$ by $g((x, k)) = k$ and $\eta : 2^{s(S_4)} \to 2^{s(S_4)}$ by

$$\eta(A) = \{(x + w, k + 1) \mid (x, k) \in A, w \in s(S_1)\}.$$

Then g and η clearly satisfy the condition of Proposition 1. Thus it follows from (13) and the proposition that

$$
\begin{aligned}
s(S_4) &= \bigcup_{n=0}^{\infty} \eta(\{(p, 0)\}) \\
&= \{(p, 0), (p + v, 1), (p + 2v, 2), (p + 3v, 3), \cdots\}.
\end{aligned}
$$

Thus $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$ contains only this cross section s with $s(S_5) = D$, and D is represented by $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_5)$.

If we set $t(S_1) = \{v, -v\}$, then

$$s(S_4) = \{(p, 0)\} \cup \{(x + v, k + 1) \mid (x, k) \in s(S_4)\} \cup \{(x - v, k + 1) \mid (x, k) \in s(S_4)\}$$

and thus

$$s(S_5) = \{\cdots, p - 3v, p - 2v, p - v, p, p + v, p + 2v, p + 3v, \cdots\}.$$

Moreover, if we set $t(S_1) = \{v, -v, u, -u\}$, then in general we get a grid points as shown in Figure 3(a).

20
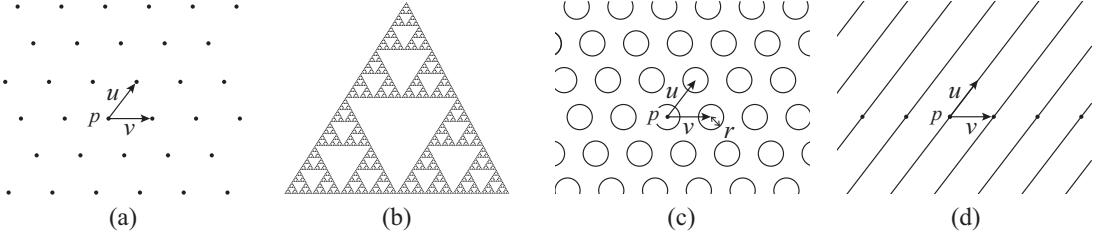
Figure 3: (a) Recursively defined points. (b) Sierpinski triangle. (c) Hierarchically defined grid of circles. (d) Hierarchically defined lines.

3.4.1 Sierpinski triangle

Consider the diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:

$$X_{(1)} \xdashrightarrow{\pi_1^{-1}} X \times X_{(2)} \xdashrightarrow{\pi_1^{-1} \circ \mathrm{sub}} V \times \mathbb{R}_{(3)} \xdashleftarrow{\pi_2^{-1}} \mathbb{R}_{(4)}$$

$$\pi_2^{-1} \uparrow \qquad\qquad \mathrm{sub}^{-1} \circ \mathrm{mult} \downarrow$$

$$X_{(5)} \xdashrightarrow{\pi_2^{-1}} X \times X_{(6)} \xdashrightarrow{\pi_1} X_{(7)}$$

Here, let $\mathscr{T} = \{S_4, S_5\}$ and $t(S_4) = \{0.5\}$, $t(S_5) = \{p\}$. Then for $s \in \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$,

$$s(S_2) = \{(x, p) \mid x \in s(S_1)\}$$
$$s(S_3) = \{(x - p, 0.5) \mid x \in s(S_1)\}$$
$$s(S_6) = \{(y, p) \mid 0.5(x - p) = y - p, x \in s(S_1)\}$$
$$s(S_7) = \{p + 0.5(x - p) \mid x \in s(S_1)\}.$$

Let us define a map $d_p : X \to X$ for $p \in X$ by $d_p(x) = p + (x - p)/2$. Then, the power map of d_p is represented by $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_1, S_7)$.

For three points $a, b,$ and c forming a triangle T in X, let us define a map $\varphi_{abc} : 2^X \to 2^X$ by $\varphi_{abc}(A) = d_a(A) \cup d_b(A) \cup d_c(A)$ for $A \subset X$. Obviously, φ_{abc} can also be represented by a diagram. Then, consider the diagram

$$\varphi'_{abc}$$

$$X_{(1)} \xdashrightarrow{\mathrm{id} \times 0} X \times \mathbb{N}_{(2)} \xdashrightarrow{\mathrm{cl} \circ \pi_1} X_{(3)}$$

Here, φ'_{abc} is φ_{abc} that increments the second component and $\mathrm{id} \times 0$ maps $A \subset X$ to $A \times \{0\} = \{(x, 0) \mid x \in A\} \subset X \times \mathbb{N}$. Let $\mathscr{T} = \{S_1\}$ and let $t(S_1) = T$. Then by Proposition 1, $s(S_2) = \bigcup_{n=0}^{\infty} \varphi'_{abc}(T \times \{0\})$. Then $s \in \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid \mathscr{T})$ gives as $s(S_3)$ the closure of $\bigcup_{n=0}^{\infty} \varphi_{abc}(T)$, which is a fractal set known as the Sierpinski triangle. See Figure 3(b).

21
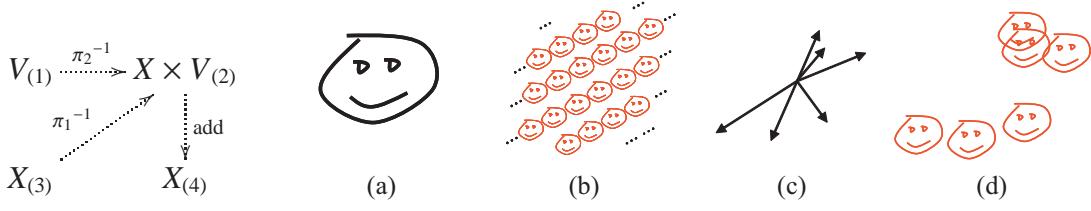
Figure 4: A diagram to make repeated patterns with arbitrary patches. (a) A patch to be B: it can be any random data. (b) Regularly repeated patches. (c) A set A of random vectors. (d) Randomly repeated patches.

3.5 Hierarchical Definition

Combining (5) and (12), we consider $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:

$$
\begin{array}{ccc}
V_{(1)} \xrightarrow{\pi_2^{-1}} X \times V \times \mathbb{N}_{(2)} & \qquad \mathbb{R}_{(3)} \xrightarrow{\mathrm{len}^{-1}} V_{(4)} \\
\pi_{13}^{-1} \Big\uparrow \Big\downarrow \varphi & \mathrm{sub}^{-1} \Big\downarrow \\
X \times \mathbb{N}_{(5)} \xrightarrow{\mathrm{id}} X \times \mathbb{N}_{(6)} \xrightarrow{\pi_1} X_{(7)} \xrightarrow{\pi_2^{-1}} X \times X_{(8)} \xrightarrow{\pi_1} X_{(9)}
\end{array}
$$

The left-half comes from (12) and produces the grid points in S_7, which is the space of center points in the right-half (5). With $\mathscr{T} = \{S_1, S_3, S_5\}$ and

$$
t(S_1) = \{v, -v, u, -u\}, \quad t(S_3) = \{r\}, \quad t(S_5) = \{(p, 0)\},
$$

$(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_9)$ represents a grid of circles as shown in Figure 3(c).

Similarly, with the diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:

$$
\begin{array}{ccc}
V_{(1)} \xrightarrow{\pi_2^{-1}} X \times V \times \mathbb{N}_{(2)} & \qquad V_{(3)} \xrightarrow{\pi_1^{-1}} V \times \mathbb{R}_{(4)} \\
\pi_{13}^{-1} \Big\uparrow \Big\downarrow \varphi & \mathrm{sub}^{-1} \circ \mathrm{mult} \Big\downarrow \\
X \times \mathbb{N}_{(5)} \xrightarrow{\mathrm{id}} X \times \mathbb{N}_{(6)} \xrightarrow{\pi_1} X_{(7)} \xrightarrow{\pi_2^{-1}} X \times X_{(8)} \xrightarrow{\pi_1} X_{(9)}
\end{array}
$$

let $\mathscr{T} = \{S_1, S_3, S_5\}$ and

$$
t(S_1) = \{v, -v\}, \quad t(S_3) = \{u\}, \quad t(S_5) = \{(p, 0)\}.
$$

Then $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_9)$ represents the lines that is parallel to u and go through the points $\{\cdots, p - 3v, p - 2v, p - v, p, p + v, p + 2v, p + 3v, \cdots\}$, as shown in Figure 3(d).

3.6 Two-part Coding

It may seem that the representation can represent only very regular and simple objects such as geometric shapes like circles and lines, except maybe by the trivial representation.

That is not the case. It can represent the regular part of the object sparsely and the random part densely, then mix them in various ways—intersection, union, hierarchically, and recursively—with any computation, in fact, as we will see. If there is regularity, it can be exploited. For example, in the diagram on the left of Figure 4, we can take any pair of subsets $A \subset V$ and $B \subset X$ and consider a partial cross section t such that $t(S_1) = A, t(S_3) = B$. This gives us, for instance, repeated patterns with arbitrary patches by setting A to be a regular grid by (12) and B to be the patch (Figure 4(b), with the patch (a)). Using an arbitrary set A as well (c), we can also have a randomly repeated patches (d). This has still much more regularity than a completely random data.

Separating the information in an object into a regular part and the remaining random part is the basic idea behind many formalisms such as Kolmogorov minimum sufficient statistic[10] and minimum description length (MDL)[11]. Our representation can also utilize this basic idea, which is how simple objects can be represented more concisely than random objects.

The representation of lines and circles by specifying points and vectors is simple; With diagrams and cross sections, such simple representation and more complex ones involving any computation can be uniformly embedded in the spaces that the objects reside. We have shown a glimpse of such integration in the grid example and the Sierpinski triangle example. In the next section, we discuss further the representation of computation by diagrams and their cross sections.

4 Computation

With diagrams and their cross sections, we can represent computation. In this section, we first examine a few examples of explicit representation of concrete computation. We then show that any Turing machine can be represented by diagrams and cross sections.

4.1 Examples

4.1.1 Factorials

Let succ $: \mathbb{N} \to \mathbb{N}$ denote the successor function and mult $: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ the multiplication. Consider the following diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:

$$\mathbb{N} \times \mathbb{N}_{(1)} \xrightarrow{\ \text{id}\ } \mathbb{N} \times \mathbb{N}_{(2)} \quad \varphi$$

where $\varphi = (\pi_1 \times \text{mult}) \circ ((\text{succ} \circ \pi_1) \times \pi_2)$ maps (n, m) to $(n + 1, m(n + 1))$. Suppose that $\mathscr{T} = \{S_1\}$ and that its cross section t is defined by

$$t(S_1) = \{(0, 1)\}.$$

Let s be a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$. Then we have

$$s(S_2) = s(S_1) \cup \varphi(s(S_2)) = \{(0, 1)\} \cup \varphi(s(S_2)). \tag{14}$$

23

We define a grading function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by $g((n, m)) = n$. Then g and φ clearly satisfy the condition of Proposition 1, and it follows from (14) that

$$s(S_2) = \bigcup_{n=0}^{\infty} \varphi(\{(0, 1)\}) = \{(0, 1), (1, 1), (2, 2), (3, 6), \cdots, (n, n!), \cdots\}. \tag{15}$$

Therefore, $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$ only contains the cross section s defined by $s(S_1) = \{(0, 1)\}$ and (15); and $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_2)$ represents the set of all pairs of natural number and its factorial.

We can extend the diagram to

$$
\begin{array}{ccccc}
\mathbb{N} \times \mathbb{N}_{(1)} & \xrightarrow{\text{id}} & \mathbb{N} \times \mathbb{N}_{(2)} & \circlearrowleft & \varphi \\
 & & \downarrow{\text{id}} & & \\
\mathbb{N}_{(3)} & \xrightarrow{\pi_1^{-1}} & \mathbb{N} \times \mathbb{N}_{(4)} & \xrightarrow{\pi_2} & \mathbb{N}_{(5)}
\end{array}
$$

Then, the (power map of the) factorial map is represented by $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_3, S_5)$.

4.1.2 Fibonacci Number

Consider the following diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:

$$
\mathbb{N}^+ \times \mathbb{N}^+_{(1)} \xrightarrow{\text{id}} \mathbb{N}^+ \times \mathbb{N}^+_{(2)} \xrightarrow{\pi_1} \mathbb{N}^+_{(3)}
$$

$$\varphi$$

where $\mathbb{N}^+$ is the set of positive integers, $\varphi = \pi_2 \times \text{add}$ with $\text{add} : \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{N}^+$ the addition. Thus $\varphi((n, m)) = (m, n + m)$. Suppose that $\mathscr{T} = \{S_1\}$ and that its cross section t is defined by $t(S_1) = \{(1, 1)\}$. Then if $s \in \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$ we have

$$s(S_2) = \{(1, 1)\} \cup \varphi(s(S_2)).$$

If $(n, m) \in s(S_2) \backslash \{(1, 1)\}$, it must be the case that $(n, m) \in \varphi(s(S_2))$ and $(m-n, n) \in s(S_2)$. If this is not in $s(S_2)$, then $(2n - m, m - n)$ is in $s(S_2)$, and so on. Since the sum of the two components decreases by this process, it cannot go on indefinitely and has to stop by reaching $(1, 1)$. Thus

$$s(S_2) = \bigcup_{n=0}^{\infty} \varphi(\{(1, 1)\}) = \{(1, 1), (1, 2), (2, 3), (3, 5), \cdots\}$$

and $s(S_3) = \{1, 1, 2, 3, 5, 8, \cdots\}$, which is the set of Fibonacci numbers.

4.1.3 Mandelbrot Set

Consider the diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:

$$\varphi \curvearrowright \mathbb{C} \times \mathbb{C} \times \mathbb{N}_{(1)} \xrightarrow{\ \pi_{12}\ } \mathbb{C} \times \mathbb{C}_{(2)} \xrightarrow{\ \mathrm{cmpl} \circ \pi_1\ } \mathbb{C}_{(3)}$$

$$\mathrm{id} \downarrow \qquad\qquad \pi_2^{-1} \uparrow$$

$$\mathbb{C} \times \mathbb{C} \times \mathbb{N}_{(4)} \qquad\qquad \mathbb{C}_{(5)}$$

Here, $\mathbb{C}$ is the set of complex numbers. The map φ is defined by

$$\varphi((c, z, k)) = (c, c + z^2, k + 1).$$

Suppose that $\mathscr{T} = \{S_4, S_5\}$ and that its cross section t is defined by

$$t(S_4) = \{(c, 0, 0) \in \mathbb{C} \times \mathbb{C} \times \mathbb{N} \mid c \in \mathbb{C}\},$$
$$t(S_5) = \{z \in \mathbb{C} \mid |z| > 2\}$$

Let s be a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$. Then we have

$$s(S_1) = \varphi(s(S_1)) \cup s(S_4) = \varphi(s(S_1)) \cup \{(c, 0, 0) \in \mathbb{C} \times \mathbb{C} \times \mathbb{N} \mid c \in \mathbb{C}\}.$$

Define $g : s(S_1) \to \mathbb{N}$ by $g((c, z, k)) = k$. Then by Proposition 1,

$$s(S_1) = \bigcup_{n=0}^{\infty} \varphi(s(S_4)) = \{(c, z_n^c, n) \in \mathbb{C} \times \mathbb{C} \times \mathbb{N} \mid c \in \mathbb{C}, n \in \mathbb{N}\},$$

where z_n^c is defined by $z_0^c = 0$ and $z_{n+1}^c = (z_n^c)^2 + c$.

Thus $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$ contains only one cross section s given by:

$$s(S_1) = \{(c, z_n^c, n) \in \mathbb{C} \times \mathbb{C} \times \mathbb{N} \mid c \in \mathbb{C}, n \in \mathbb{N}\},$$
$$s(S_2) = \{(c, z_n^c) \in \mathbb{C} \times \mathbb{C} \mid c \in \mathbb{C}, n \in \mathbb{N}, |z_n^c| > 2\},$$
$$s(S_3) = \{c \in \mathbb{C} \mid \exists n \in \mathbb{N}, |z_n^c| > 2\},$$
$$s(S_4) = \{(c, 0, 0) \mid c \in \mathbb{C}\},$$
$$s(S_5) = \{z \in \mathbb{C} \mid |z| > 2\}.$$

The Mandelbrot set is defined to be the set of complex numbers c such that z_n^c does not tend to infinity. It is known that z_n^c tends to infinity if and only if $|z_n^c| > 2$ for some $n \in \mathbb{N}$. Thus $s(S_3)$ is the Mandelbrot set.

4.1.4 Sum

Let X be a finite set and let $(a_x)_{x \in X}$ be real numbers indexed by X. Consider the following diagram $(\mathcal{S}, \mathcal{S}', \mathcal{M})$:

where the maps are defined as:

$$s : X \ni x \mapsto \{x\} \in 2^X$$

$$\cap : 2^X \times 2^X \ni (A, B) \mapsto A \cap B \in 2^X$$

$$\varphi : 2^X \times \mathbb{R} \times 2^X \times \mathbb{R} \ni (A, x, B, y) \mapsto (A \cup B, x + y) \in 2^X \times \mathbb{R}$$

Suppose that $\mathcal{T} = \{S_5, S_7\}$ and that its cross section t is defined by $t(S_5) = \{\emptyset\}$ and $t(S_7) = \{X\}$. Let s be a cross section in $\Gamma(\mathcal{S}, \mathcal{S}', \mathcal{M} \mid t)$ and assume

$$s(S_1) = \{(x, a_x) \mid x \in X\}.$$

Then we have

$$s(S_2) = \left[\{((\{x\}, a_x) \mid x \in X\} \cup \varphi(s(S_6))\right] \cap \pi_1^{-1}({}^c s(S_5))$$

$$= \{(\{x\}, a_x) \mid x \in X\} \cup \{(A \cup B, a + b) \mid (A, a, B, b) \in s(S_6), A \cup B \neq \emptyset\} \tag{16}$$

$$s(S_6) = \{(A, a, B, b) \mid (A, a), (B, b) \in s(S_2), A \cap B = \emptyset\} \tag{17}$$

Let us denote $a_A = \sum_{x \in A} a_x$ for $A \subset X, A \neq \emptyset$.

Proposition 2.

$$s(S_2) = \{(A, a_A) \mid A \subset X, A \neq \emptyset\}.$$

Proof. Define $s_n = \{(A, a) \in s(S_2) \mid |A| = n\}$ and $t_n = \{(A, a_A) \mid A \subset X, |A| = n\}$ for $n = 1, 2, \cdots, |X|$. We use an induction on n to prove $s_n = t_n$. Suppose $s_1 \ni (A, a)$. If (A, a) is not of the form $(\{x\}, a_x)$ for some $x \in X$, it is in $\varphi(s(S_6))$ and there must be $(B, b), (C, c) \in s(S_2)$ such that $A = B \cup C$ and $B \cap C = \emptyset$. However, it is impossible since $|A| = 1$ and there is no element in $s(S_2)$ of the form $(\emptyset, a)$. Thus $s_1 \subset t_n$. Since $s_1 \supset t_1$ by (16), this proves the case $n = 1$. Now, suppose $n \geq 2$ and $s_n \ni (A, a)$. Since $|A| \geq 2$, (A, a) must be in $\varphi(s(S_6))$. Then there exist $(B, b), (C, c) \in s(S_6)$ such that $A = B \cup C, B \cap C = \emptyset$, and $a = b + c$. But by the induction hypothesis $b = a_B, c = a_C$ so $a = a_B + a_C = a_A$. Thus $s_n \subset t_n$. On the other hand, suppose $(A, a_A) \in t_n$ and $x \in A$. Then $(\{x\}, a_x) \in t_1 = s_1 \subset s(S_2)$ and $(A \setminus \{x\}, a_{A \setminus \{x\}}) \in t_{n-1} = s_{n-1} \subset s(S_2)$ by the induction hypothesis. Thus $(A, a_A) = \varphi((\{x\}, a_x, A \setminus \{x\}, a_{A \setminus \{x\}})) \in s(S_2)$. Thus $(A, a_A) \in s_n$ and $s_n \supset t_n$. □

26

It follows

$$s(S_3) = \{(X, a_X)\}$$
$$s(S_4) = \{a_X\}.$$

Thus $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, S_1, S_4)$ represents the map $\Sigma_X : 2^{X \times \mathbb{R}} \to 2^{\mathbb{R}}$ that satisfies

$$\Sigma_X(\{(x, a_x) \mid x \in X\}) = \left\{ \sum_{x \in X} a_x \right\}.$$

For a general subset $A \subset X \times \mathbb{R}$, Σ_X gives

$$\Sigma_X(A) = \left\{ \sum_{x \in X} f(x) \;\middle|\; f : X \to \mathbb{R}, \text{ s.t. } \forall x \in X, (x, f(x)) \in A \right\}.$$

4.1.5 Markov Random Field

A Markov random field (MRF) consists of an undirected graph $G = (V, E)$, a finite set L of labels, and an energy function $\mathscr{E}$ on the space L^V of label assignments to vertices, or *configurations*. The energy function must be of the form:

$$\mathscr{E}(X) = \sum_{C \in \mathscr{C}} E_C(X),$$

where $\mathscr{C}$ denotes the set of cliques in G and E_C a function on L^V with the property that $E_C(X)$ depends only on values of $X \in L^V$ on C. An MRF is of the first-order if $E_C(X)$ is constant unless C is a vertex or an edge, and thus the energy can be written as:

$$\mathscr{E}(X) = \sum_{v \in V} E_1(v, X(v)) + \sum_{(u,v) \in E} E_2(u, X(u), v, X(v))$$

with $E_1 : V \times L \to \mathbb{R}$ and $E_2 : V \times L \times V \times L \to \mathbb{R}$. We assume that $E_2(u, L, v, L') = 0$ unless (u, v) is an edge. Solving an MRF involves finding the configuration X with the minimum $\mathscr{E}(X)$.

Consider the following diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:
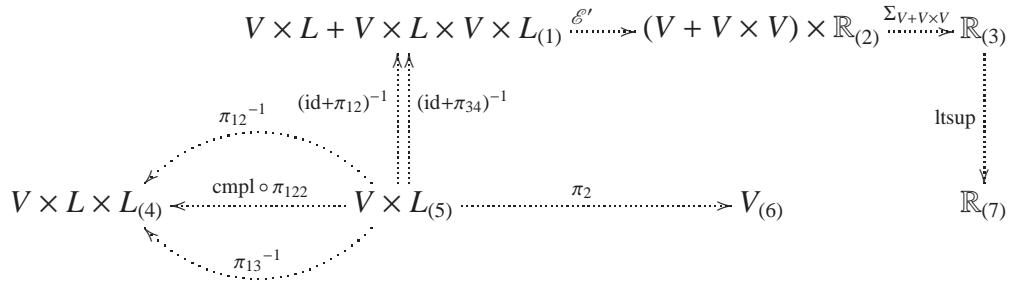
Define the maps

$$E_1' = \pi_1 \times E_1 : V \times L \to V \times \mathbb{R}$$
$$E_2' = \pi_1 \times \pi_3 \times E_2 : V \times L \times V \times L \to V \times V \times \mathbb{R}.$$

Also, define the map

$$\psi : V \times \mathbb{R} + V \times V \times \mathbb{R} \to (V + V \times V) \times \mathbb{R}$$

by

$$\psi = (\iota_1 \circ \pi_1 + \iota_2 \circ \pi_{12}) \times (\pi_2 + \pi_3).$$

Then

$$\mathscr{E}' = \psi \circ (E_1' + E_2') : V \times L + V \times L \times V \times L \to (V + V \times V) \times \mathbb{R}$$

The map $\Sigma_{V+V \times V} : (V + V \times V) \times \mathbb{R} \to \mathbb{R}$ is defined in 4.1.4. Also, define ltsup : $2^{\mathbb{R}} \to 2^{\mathbb{R}}$ by ltsup$(A) = \{x \in \mathbb{R} \mid \exists y \in A, x \le y\}$, which can be represented by

$$\mathbb{R} \xrightarrow{\pi_2^{-1}} \mathbb{R} \times \mathbb{R} \xrightarrow{\pi_1 \times \le} \mathbb{R} \times \mathbf{2} \xrightarrow{\pi_1} \mathbb{R}$$
$$\uparrow \pi_2^{-1}$$
$$\mathbf{2}$$

where the map $\le: \mathbb{R} \times \mathbb{R} \to \mathbf{2}$ maps (x, y) to 1 if $x \le y$ and 0 otherwise, and a partial cross section is defined by $s(\mathbf{2}) = \{1\}$.

Suppose that $\mathscr{T} = \{S_4, S_6\}$ and that its cross section t is defined by $t(S_4) = \emptyset, t(S_6) = V$. Let s be a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$. Then, from

$$
\begin{aligned}
s(S_4) &= \{(v, l_1, l_2) \mid (v, l_1), (v, l_2) \in s(S_5)\} \cap {}^c\{(v, l, l) \mid (v, l) \in s(S_5)\} \\
&= \{(v, l_1, l_2) \mid (v, l_1), (v, l_2) \in s(S_5), l_1 \ne l_2\} \\
&= \emptyset
\end{aligned}
$$

it follows that there is at most one element $(v, l) \in s(S_5)$ for each $v \in V$. From $s(S_6) = \pi_2(s(S_5)) = V$, there must be one for each v, thus there is exactly one element $(v, l) \in s(S_5)$ for each $v \in V$. Thus, $s(S_5)$ defines a configuration $X_s \in L^V$ by defining $X_s(v) = l$ for each $(v, l) \in s(S_5)$. Then,

$$
\begin{aligned}
s(S_1) &= \{(v, X_s(v)) \mid v \in V\} + \{(u, X_s(u), v, X_s(v))) \mid u, v \in V\} \\
s(S_2) &= \{(v, E_1(v, X_s(v)))) \mid v \in V\} + \{((u, v), E_2(u, X_s(u), v, X_s(v)))) \mid u, v \in V\} \\
s(S_3) &= \{\mathscr{E}(X_s)\} \\
s(S_7) &= \{x \in \mathbb{R} \mid x \le \mathscr{E}(X_s)\}
\end{aligned}
$$

Thus by minimizing on S_7, $\min_{S_7} \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$ gives the cross sections in $s(S_5)$ that give the configurations with the minimum energy in $s(S_3)$. Obviously, higher order MRFs can be treated in the same manner.

4.1.6 Finite Automaton

Let $(Q, \Sigma, \delta, q_0, F)$ be a finite automaton, where Q and Σ are the finite sets of states and symbols, respectively, and $\delta : \Sigma \times Q \to Q$ is the transition function, $q_0 \in Q$ is the initial state, and $F \subset Q$ is the accepting subset of Q.

Consider the diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:

$$
\begin{array}{ccccccc}
\Sigma^*_{(1)} & & \Sigma^* \times Q \times \mathbb{N}_{(2)} & \xrightarrow{\;\;\pi_{12}\;\;} & \Sigma^* \times Q_{(3)} & \xleftarrow{\;\;\pi_2^{-1}\;\;} & Q_{(4)} \\
& \pi_1^{-1} & \Big\uparrow \mathrm{id} & \delta^* & \pi_1^{-1} \Big\uparrow & & \pi_2 \\
Q \times \mathbb{N}_{(5)} & \xrightarrow{\;\pi_{23}^{-1}\;} & \Sigma^* \times Q \times \mathbb{N}_{(6)} & & \Sigma^*_{(7)} & & Q_{(8)}
\end{array}
$$

Here, Σ^* is the set of strings on Σ and

$$
\delta^* : \Sigma^* \times Q \times \mathbb{N} \ni (\tau, q, k) \mapsto \begin{cases} (\tau, q, k+1) & \text{if } \tau = \epsilon \\ (\mathrm{cdr}(\tau), \delta(\mathrm{car}(\tau), q), k+1) & \text{otherwise} \end{cases}
$$

where $\mathrm{car}(\tau) \in \Sigma$ is the first symbol in the string τ and $\mathrm{cdr}(\tau) \in \Sigma^*$ is the rest of the string.

Suppose that $\mathscr{T} = \{S_1, S_4, S_5, S_7\}$ and that its cross section t is defined by

$$
t(S_1) = \{\sigma\}, \quad t(S_4) = F, \quad t(S_5) = \{(q_0, 0)\}, \quad t(S_7) = \{\epsilon\},
$$

where $\sigma \in \Sigma^*$ is a string and ϵ denotes the empty string.

Let s be a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \,|\, t)$. Then

$$
s(S_6) = \{(\sigma, q_0, 0)\},
$$
$$
s(S_2) = s(S_6) \cup \delta^*(s(S_2)).
$$

Define $g : s(S_2) \to \mathbb{N}$ by $g((\tau, q, k)) = k$. Then by Proposition 1,

$$
s(S_2) = \bigcup_{n=0}^{\infty} \delta^*(s(S_6)) = \{(\delta^*)^n((\sigma, q_0, 0)) \mid n = 0, 1, 2, \cdots\}.
$$

Thus $s(S_2)$ is the whole execution history of the automaton, beginning with $(\sigma, q_0, 0)$. Therefore, from

$$
s(S_3) = \{(\epsilon, q) \mid (\epsilon, q, k) \in s(S_2), q \in F\},
$$
$$
s(S_8) = \{q \mid q \in F, {}^{\exists}(\epsilon, q) \in s(S_3)\},
$$

$s(S_8)$ is nonempty if and only if the automaton accepts σ.

4.2 Turing completeness

4.2.1 Turing Machine

Let $M = (Q, \Sigma, \Theta, \textvisiblespace, \delta, q_0, q_A, q_R)$ be a Turing machine, where Q is the finite set of states, Σ is the input alphabet, Θ is the finite set of internal symbols such that $\Sigma \subset \Theta$, $\textvisiblespace$ is the blank symbol in $\Theta \setminus \Sigma$, $\delta : \Theta \times Q \to \Theta \times Q \times \{-1, 1\}$ is the transition function, and $q_0, q_A, q_R \in Q$ are the initial, accept, and reject states, respectively.

Consider the following diagram $(\mathcal{S}, \mathcal{S}', \mathcal{M})$:

$$
\begin{array}{ccccc}
\Sigma^*_{(1)} & & \Theta^* \times Q \times \mathbb{N} \times \mathbb{N}_{(2)} & \xrightarrow{\;\pi_2\;} & Q_{(3)} \\[2mm]
& \pi_1^{-1} & \quad\quad \text{id} \downarrow \;\; \circlearrowright \;\; \delta^* & & \uparrow \pi_2^{-1} \\[2mm]
Q \times \mathbb{N} \times \mathbb{N}_{(4)} & \xrightarrow{\;\pi_{234}^{-1}\;} & \Theta^* \times Q \times \mathbb{N} \times \mathbb{N}_{(5)} & & Q_{(6)}
\end{array}
\tag{18}
$$

Among the components of $\Theta^* \times Q \times \mathbb{N} \times \mathbb{N} \ni (\sigma, q, n, k)$, the first three represent a configuration of the machine, with σ the content of the tape, q the state of the machine, and n the position of the head. The last component k is a step number. The map $\delta^* : \Theta^* \times Q \times \mathbb{N} \times \mathbb{N} \to \Theta^* \times Q \times \mathbb{N} \times \mathbb{N}$ is defined so that it updates the configuration of the machine by one step, as follows.

For a string σ, $\sigma[n]$ denotes the n'th symbol in the string. We use the zero-based index, so $\sigma[0]$ denotes the first symbol of σ.

First, we define

$$
\eta : \Theta^* \times Q \times \mathbb{N} \times \mathbb{N} \times (\Theta \times Q \times \{-1, 1\}) \to \Theta^* \times Q \times \mathbb{N} \times \mathbb{N}.
\tag{19}
$$

Suppose $(\sigma, q, n, k, (x, q', v)) \in \Theta^* \times Q \times \mathbb{N} \times \mathbb{N} \times (\Theta \times Q \times \{-1, 1\})$. If $q = q_A$ or $q = q_R$, $\eta(\sigma, q, n, k, (x, q', v)) = (\sigma, q, n, k + 1)$. Otherwise, $\eta(\sigma, q, n, k, (x, q', v)) = (\sigma', q', n', k + 1)$, where $n' = n + v$ unless $n + v < 0$, in which case $n' = 0$. As for σ', it is the string made from σ by (i) replacing $\sigma[n]$ by x if $0 \le n \le |\sigma| - 1$, or (ii) appending x if $n = |\sigma|$, where $|\sigma|$ denotes the length of the string σ.

Next, define $\theta : \Theta^* \times Q \times \mathbb{N} \to \Theta \times Q \times \{-1, 1\}$ by

$$
\theta(\sigma, q, n) = \begin{cases} \delta(\sigma[n], q) & \text{if } 0 \le n \le |\sigma| - 1 \\ \delta(\textvisiblespace, q) & \text{otherwise} \end{cases}
$$

for $(\sigma, q, n) \in \Theta^* \times Q \times \mathbb{N}$.

Then δ^* is defined by

$$
\delta^*(\sigma, q, n, k) = \eta(\sigma, q, n, k, \theta(\sigma, q, n)).
$$

Suppose that $\mathcal{T} = \{S_1, S_4, S_6\}$ and that its cross section t is defined by

$$
t(S_1) = \{\sigma\}, \quad t(S_4) = \{(q_0, 0, 0)\}, \quad t(S_6) = \{q_A, q_R\},
$$

where $\sigma \in \Sigma^*$.

Let s be a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid t)$. Then

$$s(S_5) = \{(\sigma, q_0, 0, 0)\},$$
$$s(S_2) = \{(\sigma, q_0, 0, 0)\} \cup \delta^*(s(S_2)).$$

Define $g : s(S_2) \to \mathbb{N}$ by $g((\tau, q, n, k)) = k$. Then by Proposition 1,

$$s(S_2) = \bigcup_{n=0}^{\infty} \delta^{*n}(s(S_5)) = \{(\delta^*)^n((\sigma, q_0, 0, 0)) \mid n = 0, 1, 2, \cdots\}.$$

Thus $s(S_2)$ is the whole execution history of the Turing machine M, beginning with $(\sigma, q_0, 0, 0)$. Therefore, if M accepts the string σ, $s(S_3) = \{q_A\}$; if it rejects σ, $s(S_3) = \{q_R\}$; and if it does not terminate, $s(S_3) = \emptyset$.

4.2.2 Nondeterministic Turing Machine

The case of nondeterministic Turing machine is similar to the deterministic case. Let $N = (Q, \Sigma, \Theta, \llcorner, \delta', q_0, q_A, q_R)$ be a nondeterministic Turing machine, where Q, Σ, Θ, $\llcorner$, q_0, q_A, and q_R are as before, and $\delta' : \Theta \times Q \to 2^{\Theta \times Q \times \{-1,1\}}$ is the nondeterministic transition function that gives the set of possible transitions.

The same diagram (18) as the deterministic case will do with just a modification of the definition of $\delta^* : 2^{\Theta^* \times Q \times \mathbb{N} \times \mathbb{N}} \to 2^{\Theta^* \times Q \times \mathbb{N} \times \mathbb{N}}$ as follows:

$$\delta^*(A) = \{\eta(\sigma, q, n, k, \tau) \mid (\sigma, q, n, k) \in A, \tau \in \theta'(\sigma, q, n)\},$$

where $\theta' : \Theta^* \times Q \times \mathbb{N} \to 2^{\Theta \times Q \times \{-1,1\}}$ is defined by

$$\theta'(\sigma, q, n) = \begin{cases} \delta'(\sigma[n], q) & \text{if } 0 \le n \le |\sigma| - 1 \\ \delta'(\llcorner, q) & \text{otherwise} \end{cases}$$

and η is as before (19).

The cross section has $s(S_9) \ni q_A$ if and only if N accepts the string σ.

5 Information Measure of Structure

We define an information measure of general subsets. The measure takes into account the structure of the set characterized by maps, which we call the *structure maps*. Thus, it is a measure of information relative to the structure maps. Essentially, the measure is the size of the smallest diagram that represents the given subset, in which the maps in the diagram can be written in terms of the structure maps.

5.1 Maps Generating Diagrams

Definition 8. Let $\mathcal{M}$ be a set of maps. The set $\langle \mathcal{M} \rangle$ of maps *generated by* $\mathcal{M}$ is defined as follows:

a) If f is in $\mathcal{M}$, or it is id, ω, or a projection map π_i, then f is in $\langle \mathcal{M} \rangle$.

b) If maps $f : X \to Y, g : Y \to Z$ are in $\langle \mathcal{M} \rangle$, then $g \circ f : X \to Z$ is also in $\langle \mathcal{M} \rangle$.

c) If maps $f_i : X \to Y_i$ ($i = 1, \cdots, n$) are in $\langle \mathcal{M} \rangle$, then the product map $f_1 \times \cdots \times f_n : X \to Y_1 \times \cdots \times Y_n$ is also in $\langle \mathcal{M} \rangle$.

Note that for any X and $x \in X$, $x : \mathbf{1} \to X$ is a map, which can be contained in the set $\mathcal{M}$ of maps in this context. When $\mathcal{M}$ contains all maps $x : \mathbf{1} \to X$, we write $X \subset \mathcal{M}$.

Definition 9. Let $\mathcal{M}$ be a set of maps. For a map f in $\langle \mathcal{M} \rangle$, its *size* $|f|_{\mathcal{M}}$ relative to $\mathcal{M}$ is defined as follows:

a) If f is in $\mathcal{M}$, or it is id, ω, or a projection map π_i, then $|f|_{\mathcal{M}} = 1$.

b) Otherwise, f is either a concatenation or a product of some maps in $\langle \mathcal{M} \rangle$. In this case, $|f|_{\mathcal{M}}$ is 1 plus the minimum of the sum of the size of component maps relative to $\mathcal{M}$, among all possible combination, i.e.,

  i) if $f \in \langle \mathcal{M} \rangle$ is a concatenation,

$$|f|_{\mathcal{M}} = 1 + \min_{\substack{g,h \in \langle \mathcal{M} \rangle \\ f = g \circ h}} (|g|_{\mathcal{M}} + |h|_{\mathcal{M}}),$$

  ii) if it is a product,

$$|f|_{\mathcal{M}} = 1 + \min_{\substack{f_1, \cdots, f_n \in \langle \mathcal{M} \rangle \\ f = f_1 \times \cdots \times f_n}} (|f_1|_{\mathcal{M}} + \cdots + |f_n|_{\mathcal{M}}),$$

Definition 10. Let $\mathcal{M}_S$ be a set of maps. A diagram $(\mathcal{S}, \mathcal{S}', \mathcal{M})$ is said to be *generated by* $\mathcal{M}_S$ if each map in $\mathcal{M}$ is either

a) the power map $f : 2^X \to 2^Y$ of a map $f : X \to Y$ in $\langle \mathcal{M}_S \rangle$ or

b) the inverse map $f^{-1} : 2^Y \to 2^X$ of a map $f : X \to Y$ in $\langle \mathcal{M}_S \rangle$.

Definition 11. Let $\mathcal{M}_S$ be a set of maps. A diagram $(\mathcal{S}, \mathcal{S}', \mathcal{M})$ is said to be *generated by* $\mathcal{M}_S$ *with complement* if each map in $\mathcal{M}$ is either

a) the power map $f : 2^X \to 2^Y$ of a map $f : X \to Y$ in $\langle \mathcal{M}_S \rangle$,

b) the inverse map $f^{-1} : 2^Y \to 2^X$ of a map $f : X \to Y$ in $\langle \mathcal{M}_S \rangle$, or

c) a complement map cmpl : $2^X \to 2^X$ for a set X in $\mathscr{S}$.

Definition 12. Let $\mathscr{M}_S$ be a set of maps and $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ a diagram generated by $\mathscr{M}_S$ with or without complement. For a map φ in $\mathscr{M}$, its size $|\varphi|_{\mathscr{M}_S}$ relative to $\mathscr{M}_S$ is defined as follows:

a) if $\varphi = \text{cmpl}$, $|\varphi|_{\mathscr{M}_S} = 1$.

b) If $\varphi = f$ or $\varphi = f^{-1}$ with $f \in \langle \mathscr{M}_S \rangle$, then $|\varphi|_{\mathscr{M}_S} = |f|_{\mathscr{M}_S}$.

In the case of the circle example (5), the diagram is generated by $\mathscr{M}_S = \{\text{len}, \text{sub}\}$ and each map in the diagram has the size 1 relative to $\mathscr{M}_S$. The diagram (6) for line is generated by $\{\text{sub}, \text{mult}\}$.

5.2 Definition

We would like to define a measure of structural information of a subset $A \subset X$ relative to a fixed set $\mathscr{M}_S$ of structure maps by considering all data $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, X)$ that represents A and such that the diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ is generated by $\mathscr{M}_S$, and then taking the minimum of the total size of the maps appearing in such diagrams. However, such a measure is not very useful without any restriction, since for any $A \subset X$ and any $\mathscr{M}_S$, such minimum is zero because of the trivial representation in which $\mathscr{S} = \mathscr{T} = \{X\}$, $\mathscr{M} = \emptyset$, and t is defined by $t(X) = A$.

To remedy this problem, we balance the data supplied by the partial cross section with their Shannon information[12].

Definition 13. Let X be a set, A its subset, and $\mathscr{M}_S$ a set of maps. Also, let $\mathscr{P}$ be a set of probability spaces, each space $Y \in \mathscr{P}$ with a probability measure P_Y. We define the structural information $I(A \,|\, \mathscr{M}_S, \mathscr{P})$ of A relative to $\mathscr{M}_S$ and $\mathscr{P}$ as the minimum of

$$\sum_{\varphi \in \mathscr{M}} |\varphi|_{\mathscr{M}_S} - \sum_{T \in \mathscr{T}} \log_2 P_T(t(T))$$

among all data $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \mathscr{T}, t, X)$ representing A such that $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ is a diagram generated by $\mathscr{M}_S$, each set in $\mathscr{T}$ is in $\mathscr{P}$, and $t(T)$ is a measurable subset of each $T \in \mathscr{T}$. If there does not exist such a diagram with which the sum is finite, we define $I(A \,|\, \mathscr{M}_S, \mathscr{P}) = \infty$. We define the measure $I^c(A \,|\, \mathscr{M}_S, \mathscr{P})$ *with complement* similarly except that we allow diagrams generated by $\mathscr{M}_S$ with complement.

Imagine that X is itself a probability space and $X \in \mathscr{P}$. Then the existence of trivial representation gives the upper bound:

$$I(A \,|\, \mathscr{M}_S, \mathscr{P}) \leq - \log_2 P_X(A).$$

33

In general, the sets in $\mathcal{T}$ can be thought of as parameter spaces, and the Definition 13 allows for the treatment of information in an ensemble of objects, just like Shannon information but taking the structure into account. However, in this paper, we focus on the information in an individual object. Thus, for the rest of this paper, we only consider the special case of $I(A \,|\, \mathcal{M}_S, \mathcal{P})$ with $\mathcal{P} = \{\mathbf{1}\}$, making $\mathbf{1}$ a probability space with the probability measure $P_{\mathbf{1}}$ defined by $P_{\mathbf{1}}(\emptyset) = 0, P_{\mathbf{1}}(\mathbf{1}) = 1$. That is, we require $\mathcal{T} = \{\mathbf{1}\}$ and $t(\mathbf{1}) = \mathbf{1}$.

Definition 14. The structural information measures $I(A \,|\, \mathcal{M}_S)$ and $I^c(A \,|\, \mathcal{M}_S)$ are defined as $I(A \,|\, \mathcal{M}_S, \mathcal{P})$ and $I^c(A \,|\, \mathcal{M}_S, \mathcal{P})$ with $\mathcal{P} = \{\mathbf{1}\}$, respectively.

For instance, the diagram for circle (5) can be modified to:

$$
\begin{array}{ccccc}
\mathbf{1}_{(1)} & \xrightarrow{\ r\ } & \mathbb{R}_{(2)} & \xrightarrow{\ \text{len}^{-1}\ } & V_{(3)} \\
 & & & & \Big\downarrow{\scriptstyle \text{sub}^{-1}} \\
\mathbf{1}_{(4)} & \xrightarrow{\ p\ } & X_{(5)} & \xrightarrow{\ \pi_2^{-1}\ } X \times X_{(6)} \xrightarrow{\ \pi_1\ } & X_{(7)}
\end{array}
$$

The constants $r \in \mathbb{R}$ and $p \in X$ are identified with the maps $r : \mathbf{1} \to \mathbb{R}$ and $p : \mathbf{1} \to X$ with values $r(0) = r \in \mathbb{R}$ and $p(0) = p \in X$ (see 2.2.) Remember that we omit the $\mathbf{1}$ in diagrams. Thus the diagram for line (6) can be modified thus:

$$
\begin{array}{ccccc}
 & \overset{v}{\underset{\searrow}{}} & & & \\
 & V_{(1)} & \xrightarrow{\ \pi_1^{-1}\ } & V \times \mathbb{R}_{(2)} & \\
 & & & \Big\downarrow{\scriptstyle \text{sub}^{-1}\circ\text{mult}} & \\
 & \overset{p}{\underset{\searrow}{}} & & & \\
 & X_{(3)} & \xrightarrow{\ \pi_2^{-1}\ } X \times X_{(4)} & \xrightarrow{\ \pi_1\ } & X_{(5)}
\end{array}
$$

The diagram is generated by $\mathcal{M}_S = \{p, v, \text{sub}, \text{mult}\}$. The map $\text{sub}^{-1} \circ \text{mult}$ is actually not in $\langle \mathcal{M}_S \rangle$. However, in general, if we have $\varphi : 2^S \to 2^T$ and $\psi : 2^T \to 2^U$, we can always think of $S \xrightarrow{\psi \circ \varphi} U$ as an abbreviation of $S \xrightarrow{\varphi} T \xrightarrow{\psi} U$.

This way, visual patterns such as shown in Figure 1 and 3, as well as explained in Section 3, can be shown to have finite information. The circle C has the upper bound of $I(C \,|\{r, p, \text{len}, \text{sub}\}) \le 6$, while the line L has $I(L \,|\{p, v, \text{sub}, \text{mult}\}) \le 7$. When we talk about Euclidean space, we should include all maps that characterize the space X and related spaces of $\mathbb{R}, V$, which would include len, sub, mult and also any constants. Thus, the proper set of maps to estimate the structural information of subsets of Euclidean space would be something like $\mathcal{M}_E = \{\text{len}, \text{sub}, \text{mult}\} \cup X \cup V \cup \mathbb{R}$. The above estimates for the circle and line cases are unchanged: $I(C \,|\mathcal{M}_E) \le 6, I(L \,|\mathcal{M}_E) \le 7$.

6 Relation to Kolmogorov Complexity

In this section, we show that the information measure defined in the previous section generalizes Kolmogorov complexity. Here, we are only concerned with binary strings; an

extension to other alphabet should be straightforward. For a Turing machine M with binary alphabet, let the partial function defined by M be denoted by the same letter. Also, let $|\sigma|$ denote the length of the string σ.

Definition 15. The Kolmogorov complexity $K_U(\sigma)$ of a binary string $\sigma \in \mathbf{2}^*$ with respect to a universal Turing machine U is defined as

$$K_U(\sigma) = \min_{p \in \mathbf{2}^*, U(p) = \sigma} |p|.$$

6.1 Emulating Turing Machines

It turns out that any Turing machine can be emulated by a diagram generated by $\mathscr{M}_{\mathbb{N}} = \{0, \text{succ}\}$, where $\text{succ} : \mathbb{N} \to \mathbb{N}$ is the successor function and $0 : \mathbf{1} \to \mathbb{N}$, if we represent a binary string $\sigma \in \mathbf{2}^*$ by the subset

$$\bar{\sigma} = \{(i, \sigma[i]) \mid i = 0, 1, \cdots, |\sigma| - 1\} \cup \{(|\sigma|, 0), (|\sigma|, 1)\} \subset \mathbb{N} \times \mathbb{N}.$$

Theorem 1. *Let M be a Turing machine with binary alphabet. Then there exists a finite diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ generated by $\mathscr{M}_{\mathbb{N}} = \{0, \text{succ}\}$ and two sets $S = \mathbb{N} \times \mathbb{N}$ and $T = \mathbb{N} \times \mathbb{N}$ in $\mathscr{S}$ such that $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \{\mathbf{1}\}, s_1, S, T)$ represents the map $\bar{M} : \mathbf{2}^{\mathbb{N} \times \mathbb{N}} \to \mathbf{2}^{\mathbb{N} \times \mathbb{N}}$ that maps $\bar{\sigma}$ to $\bar{\tau}$ if M accepts σ leaving a string τ on its tape, or to $\emptyset$ if M rejects σ or does not terminate with input σ.*

Proof. Let $M = (Q, \mathbf{2}, \Theta, \llcorner, \delta, q_0, q_A, q_R)$, where Q is the finite set of states, Θ is the finite set of internal symbols such that $\mathbf{2} \subset \Theta$, $\llcorner \in \Theta$ is the blank symbol, $\delta : \Theta \times Q \to \Theta \times Q \times \mathbf{2}$ is the transition function, and $q_0, q_A, q_R \in Q$ are the initial, accept, and reject states, respectively. We define $\tilde{Q} = Q + \{q_e\}$ and $\tilde{\Theta} = \Theta + \{\square\}$. Consider the following diagram $(\mathscr{S}, \mathscr{S}', \mathscr{M})$:

$$\mathbb{N} \times \mathbb{N}_{(1)} \xdashrightarrow{\text{init}} \mathbb{N} \times \tilde{\Theta} \times \tilde{Q} \times \mathbb{N}_{(2)} \xdashrightarrow{\text{id}} \mathbb{N} \times \tilde{\Theta} \times \tilde{Q} \times \mathbb{N}_{(3)} \xdashrightarrow{\theta \circ \pi_{12}} \mathbb{N} \times \mathbb{N}_{(4)} \qquad (20)$$

with loops δ^* at (2) and $\pi_3^{-1} \circ q_A$ from (3) to (4).

Here, S_2 is meant to represent the whole execution history of the Turing machine M.

Let us assume that σ is a binary string and s is a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid s_1)$ with $s(S_1) = \bar{\sigma}$. For $k \in \mathbb{N}$, let us call the set $R_k = \{(i, x, q, k) \in s(S_2) \mid (i, x, q) \in \mathbb{N} \times \tilde{\Theta} \times \tilde{Q}\}$ the k'th row. We call the R_k *proper* if the following condition holds:

$$R_k = R_k^1 + R_k^2;$$
$$R_k^1 = \{(i, x_{ik}, q_e, k) \mid i = 0, \cdots, l_k\}, x_{l_k k} = \square, x_{ik} \neq \square \text{ for } i < l_k;$$
$$R_k^2 = \{(i, x_{ik}, q_k, k) \mid i = m_k, \cdots, n_k\}, q_k \neq q_e, 0 \leq m_k \leq n_k \leq l_k; \text{ if } q_k \neq q_A \text{ then } m_k = n_k.$$

In a proper row, x_{ik} represents the i'th symbol on the tape at k'th step in execution of M, except for the last entry in the row, which is $\square$. When $q_k \neq q_A$, the single element

35

$(m_k, x_{m_k k}, q_k, k)$ in R_k^2 indicates that M is in state q_k and its head is at position m_k. When $q_k = q_A$ it means that the machine is in the accept state.

The map init sends the subset $\bar{\sigma}$ to

$$\text{init}(\bar{\sigma}) = \{(i, \sigma[i], q_e, 0) \mid i = 0, \cdots, |\sigma| - 1\} \cup \{(|\sigma|, \Box, q_e, 0)\} \cup \{(0, \sigma[0], q_0, 0)\}, \quad (21)$$

setting up the 0'th row properly.

The map $\delta^* : 2^{\mathbb{N} \times \tilde{\Theta} \times \tilde{Q} \times \mathbb{N}} \to 2^{\mathbb{N} \times \tilde{\Theta} \times \tilde{Q} \times \mathbb{N}}$ maps R_k to R_{k+1} with the following properties:

1. If R_k is proper and $q_k \neq q_A$, δ^* updates the configuration by emulating M, i.e., R_k^1 is copied to R_{k+1}^1 except that $x_{m_k k}$ is mapped to $x_{m_k k+1}$ according to M's tape rewriting, and that the single element $(m_k, x_{m_k k}, q_k, k)$ of R_k^2 is mapped to $(m_{k+1}, x_{m_{k+1} k}, q_{k+1}, k+1)$ in R_{k+1}^2, where m_{k+1} is the position of the head at step $k + 1$.

2. If k'th row is proper and $q_k = q_A$, R_k^1 is copied to R_{k+1}^1 and R_k^2 is expanded. That is, if

$$R_k^2 = \{(i, x_{ik}, q_A, k) \mid i = m_k, \cdots, n_k\},$$

then

$$R_{k+1}^2 = \{(i, x_{ik}, q_A, k) \mid i = \max(0, m_k - 1), \cdots, \min(l_k, n_k + 1)\}.$$

In Lemma 6, we show that such a δ^* can be represented using a subdiagram. If we define a grading map $g : \mathbb{N} \times \tilde{\Theta} \times \tilde{Q} \times \mathbb{N} \to \mathbb{N}$ by $g((i, x, q, k)) = k$, it is clear from the proof of the lemma that δ^* satisfies the requirement of Proposition 1, which is

$$\delta^*(S_i) \subset S_{i+1}, \qquad \delta^*(S) = \bigcup_{n=0}^{\infty} \delta^*(S_n),$$

where $S = \mathbb{N} \times \tilde{\Theta} \times \tilde{Q} \times \mathbb{N}$ and $S_i = g^{-1}(i)$. Thus from the proposition we have

$$s(S_2) = \bigcup_{k=0}^{\infty} \delta^{*k}(\text{init}(s(S_1)))$$

We then have the following:

$$s(S_3) = \{(i, x, q, k) \in s(S_2) \mid q = q_A\}.$$

If M reaches the accept state at k'th step, R_k^2 would have one element with the accept state q_A. The rows after that would have increasingly more q_A without changing the symbols, until $R_{k'}^2 = \{(i, x_{ik}, q_A, k') \mid i = 0, \cdots, l_k\}$. The map $\theta : 2^{\mathbb{N} \times \tilde{\Theta}} \to 2^{\mathbb{N} \times \mathbb{N}}$ just fixes the termination of the string, removing the blank symbols that might be left at the end of the string and properly terminating:

$$\theta(A) = \{(i, x) \mid (i, x), (i + 1, y) \in A, x, y \in \mathbf{2}\} \cup \{(0, 0), (0, 1) \mid (0, y) \in A, y \in B\} \cup$$
$$\{(i, x), (i + 1, 0), (i + 1, 1) \mid (i, x), (i + 1, y) \in A, x \in \mathbf{2}, y \in B\}, \qquad (22)$$

36

where $B = \{\llcorner, \square\}$. The second part of the RHS is to take care of the case when the string is empty. Thus, $s(S_4)$ contains the string that remains when M reaches the accept state if it does, and is empty if it does not.

Without loss of generality, we can assume that $Q = \mathbf{n}, q_e = n$ and $\Theta = \mathbf{m}, \square = m$. Remember, in our notation, the finite subset $\{0, 1, \cdots, n - 1\}$ of $\mathbb{N}$ is denoted by $\mathbf{n}$. Thus, constant maps such as q_e, q_0 are natural number constants. Any natural number constant map k can be made from 0 and succ as $\mathrm{succ}^k \circ 0$.

It remains to prove that the maps init, θ, and δ^* can be represented by a diagram generated by $\mathscr{M}_{\mathbb{N}}$, which is done in the following lemmas. $\square$

Lemma 1. *The map* init $: 2^{\mathbb{N} \times \mathbb{N}} \to 2^{\mathbb{N} \times \tilde{\Theta} \times \tilde{Q} \times \mathbb{N}}$ *that satisfies* (21) *can be represented by a diagram generated by* $\mathscr{M}_{\mathbb{N}}$.

Proof. Consider the diagram
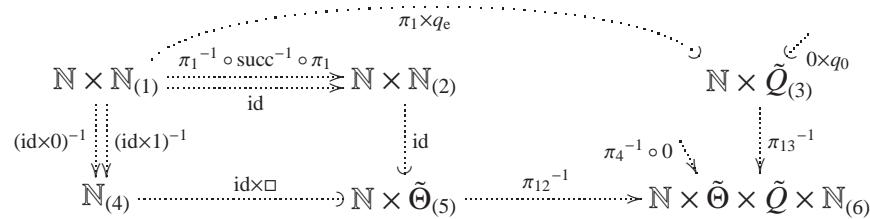
Let σ be a binary string and s a cross section of the diagram such that $s(S_1) = \bar{\sigma} = \{(i, \sigma[i]) \mid i = 0, 1, \cdots, |\sigma| - 1\} \cup \{(|\sigma|, 0), (|\sigma|, 1)\}$. Then

$$s(S_2) = \{(i, x) \mid (i + 1, x') \in s(S_1), x' \in \mathbf{2}, (i, x) \in s(S_1)\} = \{(i, \sigma[i]) \mid i = 0, \cdots, |\sigma| - 1\},$$

$$s(S_3) = \{(i, q_e) \mid (i, x) \in s(S_1), x \in \mathbf{2}\} \cup \{(0, q_0)\} = \{(i, q_e) \mid i = 0, \cdots, |\sigma|\} \cup \{(0, q_0)\},$$

$$s(S_4) = \{i \in \mathbb{N} \mid (i, 0), (i, 1) \in s(S_1)\} = \{|\sigma|\},$$

$$s(S_5) = \{(i, \sigma[i]) \mid i = 0, 1, \cdots, |\sigma| - 1\} \cup \{(|\sigma|, \square)\},$$

$$s(S_6) = \{(i, \sigma[i], q_e, 0) \mid i = 0, \cdots, |\sigma| - 1\} \cup \{(|\sigma|, \square, q_e, 0)\} \cup \{(0, \sigma[0], q_0, 0)\}.$$

Thus, $s(S_6)$ is the result of applying init to $s(S_1)$. $\square$

Lemma 2. *The map* $\theta : 2^{\mathbb{N} \times \tilde{\Theta}} \to 2^{\mathbb{N} \times \mathbb{N}}$ *that satisfies* (22) *can be represented by a diagram generated by* $\mathscr{M}_{\mathbb{N}}$.

Proof. Consider the diagram
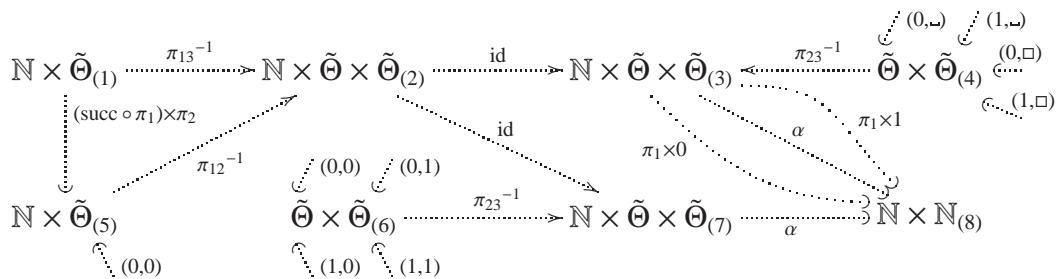
Here, $\alpha = ((\text{succ} \circ \pi_1) \times \pi_2)^{-1} \circ \pi_{12}$. Then, for a cross section s of the diagram with $s(S_1) = A$,

$$s(S_5) = \{(i + 1, x) \mid (i, x) \in A\} \cup \{(0, 0)\},$$
$$s(S_2) = \{(i + 1, x, y) \mid (i, x), (i + 1, y) \in A\} \cup \{(0, 0, y) \mid (0, y) \in A\},$$
$$s(S_4) = \{(x, y) \mid x \in \mathbf{2}, y \in B\},$$
$$s(S_3) = \{(i, x, y) \in s(S_2) \mid x \in \mathbf{2}, y \in B\}$$
$$= \{(i + 1, x, y) \mid (i, x), (i + 1, y) \in A, x \in \mathbf{2}, y \in B\} \cup \{(0, 0, y) \mid (0, y) \in A, y \in B\},$$
$$s(S_6) = \{(x, y) \mid x, y \in \mathbf{2}\},$$
$$s(S_7) = \{(i, x, y) \in s(S_2) \mid x, y \in \mathbf{2}\}$$
$$= \{(i + 1, x, y) \mid (i, x), (i + 1, y) \in A, x, y \in \mathbf{2}\} \cup \{(0, 0, y) \mid (0, y) \in A, y \in \mathbf{2}\}$$
$$s(S_8) = \{(i, x), (i + 1, 0), (i + 1, 1) \mid (i, x), (i + 1, y) \in A, x \in \mathbf{2}, y \in B\} \cup$$
$$\{(0, 0), (0, 1) \mid (0, y) \in A, y \in B\} \cup \{(i, x) \mid (i, x), (i + 1, y) \in A, x, y \in \mathbf{2}\},$$

where $B = \{\llcorner, \square\}$. Thus, $s(S_8)$ is the result of applying θ to $s(S_1)$. □

Lemma 3. *Let δ be a map from $\mathbf{m} \times \mathbf{n}$ to $\mathbf{m} \times \mathbf{n} \times \mathbf{2}$. Then δ can be represented by a diagram generated by $\mathscr{M}_{\mathbb{N}}$.*

Proof. Let k be the larger of n and m. The diagram has three parts: $S_{A,i} = \mathbb{N}$ for $i = 0, \cdots, k - 1$; $S_{B,i,j} = \mathbf{m} \times \mathbf{n} \times \mathbf{m} \times \mathbf{n} \times \mathbf{2}$ for $(i, j) \in \mathbf{m} \times \mathbf{n}$; and $S_{C,0} = \mathbf{m} \times \mathbf{n}$, $S_{C,1} = \mathbf{m} \times \mathbf{n} \times \mathbf{2}$. The first part is as follows:

$$\mathbb{N}_{(A,0)} \xrightarrow{\text{succ}} \mathbb{N}_{(A,1)} \xrightarrow{\text{succ}} \cdots \xrightarrow{\text{succ}} \mathbb{N}_{(A,k-1)}$$

with a 0 pointing to $\mathbb{N}_{(A,0)}$.

This part has no incoming arrows from other parts. Thus, any cross section s has $s(S_{A,i}) = \{i\}$ for $i = 0, 1, \cdots, k - 1$.

The other parts are organized thus:

with maps, for each $(i, j) \in \mathbf{m} \times \mathbf{n}$, as follows:

38

where $\delta(i, j) = (\delta_1(i, j), \delta_2(i, j), \delta_3(i, j))$.

Let s be a cross section of this diagram. Then for each $(i, j) \in \mathbf{m} \times \mathbf{n}$,

$$
\begin{aligned}
s(S_{\mathrm{B},i,j}) &= \pi_{12}^{-1}(s(S_{\mathrm{C},0})) \cap \pi_1^{-1}(s(S_{\mathrm{A},i})) \cap \pi_2^{-1}(s(S_{\mathrm{A},j})) \\
&\qquad \cap \pi_3^{-1}(s(S_{\mathrm{A},\delta_1(i,j)})) \cap \pi_4^{-1}(s(S_{\mathrm{A},\delta_2(i,j)})) \cap \pi_5^{-1}(s(S_{\mathrm{A},\delta_3(i,j)})) \\
&= \{(i, j, \delta_1(i, j), \delta_2(i, j), \delta_3(i, j)) \in \mathbf{m} \times \mathbf{n} \times \mathbf{m} \times \mathbf{n} \times \mathbf{2} \mid (i, j) \in s(S_{\mathrm{C},0})\} \\
&= \begin{cases} \{(i, j, \delta_1(i, j), \delta_2(i, j), \delta_3(i, j))\} & \text{if } (i, j) \in s(S_{\mathrm{C},0}) \\ \emptyset & \text{if } (i, j) \notin s(S_{\mathrm{C},0}) \end{cases}
\end{aligned}
$$

and thus

$$
\begin{aligned}
s(S_{\mathrm{C},1}) &= \bigcup_{(i,j) \in s(S_{\mathrm{C},0})} \pi_{345}(s(S_{\mathrm{B},i,j})) \\
&= \{\delta(i, j) \mid (i, j) \in s(S_{\mathrm{C},0})\} = \delta(s(S_{\mathrm{C},0})).
\end{aligned}
$$

$\square$

Lemma 4. *The map* $\psi : \mathbf{2}^{\mathbb{N} \times \tilde{\Theta} \times Q \times \mathbf{2} \times \mathbb{N}} \to \mathbf{2}^{\mathbb{N} \times Q \times \mathbb{N}}$ *defined by*

$$
\begin{aligned}
\psi(A) &= \{(0, q, k) \mid (0, x, q, 0, k) \in A\} \\
&\quad \cup \{(i + 1, q, k) \mid (i, x, q, 1, k) \in A\} \cup \{(i, q, k) \mid (i + 1, x, q, 0, k) \in A\}
\end{aligned}
$$

can be represented by a diagram generated by $\mathscr{M}_{\mathbb{N}}$.

Proof. Consider the diagram

Then we have

$$
\begin{aligned}
s(S_2) &= \{(i, q, v, k) \mid (i, x, q, v, k) \in s(S_1)\}, \\
s(S_3) &= \{(i, q, k) \mid (i, x, q, 1, k) \in s(S_1)\}, \\
s(S_5) &= \{(q, k) \mid (0, x, q, 0, k) \in s(S_1)\}, \\
s(S_4) &= \{(i + 1, q, k) \mid (i, x, q, 1, k) \in s(S_1)\} \cup \{(i, q, k) \mid (i + 1, x, q, 0, k) \in s(S_1)\} \\
&\quad \cup \{(0, q, k) \mid (0, x, q, 0, k) \in s(S_1)\}.
\end{aligned}
$$

Thus we have $s(S_4) = \psi(s(S_1))$. $\square$

Lemma 5. *The map $\chi : 2^{\mathbb{N}\times\mathbb{N}} \to 2^{\mathbb{N}\times\mathbb{N}}$ such that*

$$\chi(\{(i,k)\}) = \{(j,k) \mid j \in \mathbb{N}, j \neq i\}$$

can be represented by a diagram generated by $\mathscr{M}_{\mathbb{N}}$.

Proof. Consider the diagram

Here, $s_1 = (\text{succ} \circ \pi_1) \times \pi_2$ and $s_2 = ((\text{succ} \circ \pi_1) \times \pi_2)^{-1}$. Then if $s(S_1) = \{(i,k)\}$,

$$
\begin{aligned}
s(S_2) &= \{(j,k) \mid j = i+1, i+2, \cdots\}, \\
s(S_3) &= \{(j,k) \mid j = i-1, i-2, \cdots, 0\}, \\
s(S_4) &= \{(j,k) \mid j \in \mathbb{N}, j \neq i\}.
\end{aligned}
$$

$\square$

Lemma 6. *The map $\delta^* : 2^{\mathbb{N}\times\tilde{\Theta}\times\tilde{Q}\times\mathbb{N}} \to 2^{\mathbb{N}\times\tilde{\Theta}\times\tilde{Q}\times\mathbb{N}}$ with the properties in the proof of the theorem can be represented by a diagram generated by $\mathscr{M}_{\mathbb{N}}$.*

Proof. Consider the diagram:

For $k \in \mathbb{N}$, let us assume that $s(S_1)$'s k'th row R_k is proper. We see that $s(S_2)$ contains the single element $(i, x, q, k), q \neq q_A$ in R_k^2 and $s(S_4)$ the rest, with the step number k incremented. Note that $\{q_e, q_A\}$ and $\tilde{Q} \setminus \{q_e, q_A\}$ in the diagram denotes the constant union maps $\mathbf{1} \to \tilde{Q}$ whose images are $\{q_e, q_A\}$ and $\tilde{Q} \setminus \{q_e, q_A\}$.

40

The map $\chi \circ \pi_{14}$ sends to

$$s(S_5) = \bigcup_{(i,x,q,k) \in s(S_2)} \{(j,k) \mid j \neq i\}.$$

Thus, $s(S_5)$ contains exactly the position-step pair (i,k) such that there is no element (i,x,q,k) in $s(S_2)$. The element (i,x,q,k) in R_k^2 indicates that the head is at position i on step k. Thus $s(S_5)$ contains the position-step pair everywhere except where the head is, and $s(S_8)$ contains the symbols that are not under the head.

The symbol-state pair (x,q) from (i,x,q,k) in R_k^2 is fed to δ and the result is

$$s(S_3) = \{(i,x',q',v,k) \mid (i,x,q,k) \in s(S_2), \delta((x,q)) = (x',q',v)\},$$

which contains the symbol x' to replace the one at the current position i, the new state q', and the direction v to move the head. The new symbol is sent to $s(S_6)$ where it is combined with the symbols that are not changed, which are contained in $s(S_8)$. The direction to move the head is indicated by v: if $v = 0$, the head is to be moved to the left, and if $v = 1$ to the right. The map ψ in Lemma 4 is defined so that the position is moved accordingly, while also taking care of the case $(i,v) = (0,0)$, when the head is indicated to move to the left and the position is 0, and therefore has to stay at 0. Also, without loss of generality, we can assume that δ satisfies $\delta((x,q_R)) = (x,q_R,0)$, which makes the simulated head to move to the left without changing any symbol after the rejection state is reached, until it reaches the leftmost position, where it stays.

Coming back to $s(S_4)$,

$$s(S_7) = \{(i,q_A,k) \mid (i,x,q_A,k) \in s(S_4)\},$$

takes position-step pairs with state q_A. The map $\phi : 2^{\mathbb{N} \times \tilde{Q} \times \mathbb{N}} \to 2^{\mathbb{N} \times \tilde{Q} \times \mathbb{N}}$ is defined so that

$$\begin{aligned}
s(S_{10}) = \phi(s(S_7)) = &\{(i,q_A,k) \mid (i+1,q_A,k) \in s(S_7)\} \\
&\cup \{(i,q_A,k) \mid (i,q_A,k) \in s(S_7)\} \\
&\cup \{(i+1,q_A,k) \mid (i,q_A,k) \in s(S_7)\},
\end{aligned}$$

which propagates the positions with accept state q_A. This can be represented by

Now, the position-state-step triples are combined in $s(S_{11})$:

$$s(S_{11}) = \psi(s(S_3)) \cup s(S_{10}) \cup \{(i,q_e,k) \mid i,k \in \mathbb{N}\},$$

where the last part ensures that the state q_e is in R_k^1 everywhere there is a symbol.

The states and the symbols are combined in $s(S_9)$: which is combined with the symbols to make

$$s(S_9) = \{(i, x, q, k) \mid (i, x, k) \in s(S_6), (i, q, k) \in s(S_{11})\}.$$

Finally, the map $\eta : 2^{\mathbb{N} \times \tilde{\Theta} \times \tilde{Q} \times \mathbb{N}} \to 2^{\mathbb{N} \times \tilde{\Theta} \times \tilde{Q} \times \mathbb{N}}$ takes care of the case when the head moves past the right end of the string:

$$\eta((i, \square, q, k)) = \{(i, \llcorner, q, k), (i + 1, \square, q_e, k)\}, \quad (q \neq q_e, q \neq q_A),$$
$$\eta((i, x, q, k)) = \{(i, x, q, k)\} \quad \text{otherwise.}$$

This can be represented by:

□

Theorem 2. *For any universal Turing machine U, there exists a constant $c_U \in \mathbb{N}$ such that*

$$I(\bar{\sigma} \mid \mathcal{M}_{\mathbb{N}}) \leq 6K_U(\sigma) + c_U$$

for any binary string $\sigma \in 2^*$.

Proof. By Theorem 1, there exists a diagram $(\mathcal{S}, \mathcal{S}', \mathcal{M})$ generated by $\mathcal{M}_{\mathbb{N}}$ that emulates U. By definition, there exists $p \in 2^*$ such that $U(p) = \sigma$ and $|p| = K_U(\sigma)$. There are sets $S = \mathbb{N} \times \mathbb{N}$ and $T = \mathbb{N} \times \mathbb{N}$ in $\mathcal{S}$, as S_1 and S_4 in (20), such that $s(T) = \bar{\sigma}$ if $s(S) = \bar{p}$, for any cross section s of $(\mathcal{S}, \mathcal{S}', \mathcal{M} \mid s_1)$. Add to the diagram the following:

with maps from $S_{A,i}$ to S:

$$\mathbb{N}_{(A,i)} \xdashrightarrow{\text{id} \times p[i]} \mathbb{N} \times \mathbb{N}$$

for $i = 0, 1, \ldots, |p| - 1$, where $\text{id} \times p[i] : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ denotes the map $\text{id} \times 0$ if $p[i] = 0$ and $\text{id} \times (\text{succ} \circ 0)$ if $p[i] = 1$. Finally, we add maps

$$\mathbb{N}_{(A,|p|)} \xdashrightarrow[\text{id} \times 1]{\text{id} \times 0} \mathbb{N} \times \mathbb{N}$$

42

from $S_{A,|p|}$ to S. Then any cross section s of the resulting diagram satisfies $s(S) = \bar{p}$, and thus $s(T) = \bar{\sigma}$.

Maps succ and id $\times p[i]$ is added for each symbol in p. The size of succ is 1 and that of id $\times$ (succ $\circ$ 0) is 5. Thus the size of the added diagram is at most $6|p|$ + const. Since the only part of the diagram that depends on σ is the part added here, the existence of this diagram proves the theorem. $\qquad\square$

Corollary 1. *For any binary string $\sigma \in \mathbf{2}^*$, $I(\bar{\sigma} | \mathcal{M}_{\mathbb{N}})$ is bounded from above by*

$$c \cdot \min_{\substack{M:\mathrm{TM}, p \in \mathbf{2}^* \\ M(p)=\sigma}} (|p| + |\Theta_M||Q_M|),$$

where c is a constant and Θ_M and Q_M are the sets of internal symbols and states of Turing machine M, respectively.

Proof. Use the same construction as in the proof of Theorem 2 with any (possibly non-universal) Turing machine. It can be seen from the proof of the theorem and Lemma 3 that the size of the diagram emulating a Turing machine M is $O(|\Theta_M||Q_M|)$. $\qquad\square$

6.2 Reading Cross Section

Conversely, if a finite diagram generated by $\mathcal{M}_{\mathbb{N}}$ can represent $\bar{\sigma}$ for some binary string σ, there is a Turing machine that produces σ and terminate.

Let X be a set of Boolean variables, i.e., variables that take values in $\mathbf{2}$. We use the standard notion like logical AND ($\wedge$) and OR ($\vee$), treating 1 as true and 0 as false. By an assignment to X, we mean a map $f : X \to \mathbf{2}$, which assigns 0 or 1 to each variable in X. If $Y \subset X$, there is a natural map $\mathbf{2}^X \to \mathbf{2}^Y$ by restriction that maps an assignment to X to an assignment to Y. A logical constraint χ on a set X of Boolean variables is a map $\chi : \mathbf{2}^X \to \mathbf{2}$, which is said to be satisfied by an assignment $f \in \mathbf{2}^X$ to X if $\chi(f) = 1$. For a logical constraint χ on X, an assignment g to a subset Y of X is said to satisfy χ if every assignment that restricts to g satisfies it. For a set C of constraints on X, we say an assignment to X satisfies C if it satisfies all the constraints in C.

Theorem 3. *Let $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ be a finite diagram generated by $\mathcal{M}_{\mathbb{N}}$ such that the data $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \{\mathbf{1}\}, s_1, S)$ represents $\bar{\sigma} \subset S = \mathbb{N} \times \mathbb{N}$ for a binary string σ, where s_1 is the cross section of $\{\mathbf{1}\}$ defined by $s_1(\mathbf{1}) = \mathbf{1}$. Then there exists a Turing machine that takes an encoded description of any such diagram and produces σ. Therefore, for any universal Turing machine U, there exist constants $d_U, e_U \in \mathbb{N}$ such that*

$$K_U(\sigma) \le d_U I(\bar{\sigma} | \mathcal{M}_{\mathbb{N}}) + e_U$$

for any binary string $\sigma \in \mathbf{2}^*$.

Proof. For each element t of each set T in $\mathscr{S}$, we define a Boolean variable x_t. Also, for each set T in $\mathscr{S}$, each map $\varphi \in \text{in}(T)$, and each element t of T, we define a Boolean variable y_t^φ. Let us denote the set of the variables by X. The variable x_t is for indicating if t is in the cross section, and y_t^φ is to indicate if t is in the image of the cross section by φ. We define a set C of constraints on X to establish a one-to-one correspondence between the cross sections in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid s_1)$ and the assignments to X satisfying C. We let C contain the following constraints:

i) For each map $\varphi \in \mathscr{M}$:

    a) If φ is the power map $f : 2^U \to 2^T$ of $f : U \to T$, C contains the constraint $y_t^\varphi = 0$ for each $t \in T \setminus f(U)$ and, for each $t \in f(U)$,

$$y_t^\varphi = \bigvee_{u \in f^{-1}(t)} x_u.$$

    b) If φ is the inverse map $f^{-1} : 2^U \to 2^T$ of $f : T \to U$, C contains the constraint

$$y_t^\varphi = x_{f(t)}$$

    for each $t \in T$.

ii) For each set T in $\mathscr{S} \setminus \mathscr{S}'$ such that $\text{in}(T) \neq \emptyset$ and each t in T, C contains the constraint

$$x_t = \bigwedge_{\varphi \in \text{in}(T)} y_t^\varphi.$$

iii) For each set T in $\mathscr{S}'$ such that $\text{in}(T) \neq \emptyset$ and each t in T, C contains the constraint

$$x_t = \bigvee_{\varphi \in \text{in}(T)} y_t^\varphi.$$

iv) For the variable x_0 corresponding to the element $0 \in \mathbf{1}$ that appears in the data $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \{\mathbf{1}\}, s_1, S)$, C contains the constraint $x_0 = 1$.

Note that for each variable $x \in X$, there is exactly one constraint in C with that variable on the LHS. We call it χ_x. Note also that any conjunction in C has only a finite number of variables and that there are only countably many variables in X. We fix a one-to-one correspondence $v : X \to \mathbb{N}$ for later use.

For a cross section s of $\mathscr{S}$, we define an assignment g_s to X by defining $g_s(x_t) = 1$ if and only if $t \in s(T)$ for each $T \in \mathscr{S}, t \in T$ and $g_s(y_t^\varphi) = 1$ if and only if $t \in \varphi(s(\text{dm}(\varphi)))$. Then g_s satisfies C if and only if s is a cross section in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid s_1)$, as follows.

First, suppose that g_s satisfies C. If $T \in \mathscr{S} \setminus \mathscr{S}'$, then $g_s(x_t) = \bigwedge_{\varphi \in \text{in}(T)} g_s(y_t^\varphi)$ and thus $t \in s(T)$ (i.e., $g_s(x_t) = 1$) if and only if $t \in \bigcap_{\varphi \in \text{in}(T)} \varphi(s(\text{dm}(\varphi)))$. If $T \in \mathscr{S}'$, then $g_s(x_t) =$

$\bigvee_{\varphi \text{in}(T)} g_s(y_t^\varphi)$; thus $t \in s(T)$ (i.e., $g_s(x_t) = 1$) if and only if $t \in \bigcup_{\varphi \text{in}(T)} \varphi(s(\text{dm}(\varphi)))$. This proves that s is in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M})$. Finally, the constraint $g_s(x_0) = 1$ for the variable x_0 corresponding to the element $0 \in \mathbf{1}$ ensures that $s(\mathbf{1}) = \mathbf{1}$, and thus s is in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid s_1)$. Conversely, suppose that s is in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid s_1)$. If $T \in \mathscr{S} \setminus \mathscr{S}'$, $g_s(x_t) = 1$ (i.e., $t \in s(T)$) if and only if $t \in \bigcap_{\varphi \text{in}(T)} \varphi(s(\text{dm}(\varphi)))$; since $g_s(y_t^\varphi) = 1$ if and only if $t \in \varphi(s(\text{dm}(\varphi)))$, $g_s(x_t) = \bigwedge_{\varphi \text{in}(T)} g_s(y_t^\varphi)$. If $T \in \mathscr{S}'$, $g_s(x_t) = 1$ (i.e., $t \in s(T)$) if and only if $t \in \bigcup_{\varphi \text{in}(T)} \varphi(s(\text{dm}(\varphi)))$; thus $g_s(x_t) = \bigvee_{\varphi \text{in}(T)} g_s(y_t^\varphi)$. Finally, $g_s(x_0) = 1$ is satisfied since $s(\mathbf{1}) = \mathbf{1}$.

We define a subset X_i of X for $i \in \mathbb{N}$ by

$$X_0 = \{x_0\},$$
$$X_{i+1} = X_i \cup \{x \in X \mid x \text{ is forced by } X_i\}.$$

Here, a variable $x \in X$ is said to be forced by a subset Y of X either if χ_x is a disjunction and at least one variable on its RHS is in Y or if χ_x is a conjunction and all the variables on its RHS is in Y. Note that, if an assignment g to X satisfies C, $g(x_0) = 1$ and, by following the definition, any variable x in X_i has $g(x) = 1$.

We define a function τ on X:

$$\tau(x) = \begin{cases} i & \text{if } x \in X_i \setminus X_{i-1} \\ \infty & \text{otherwise} \end{cases}$$

and using τ, we define an assignment h to X by:

$$h(x) = 1 \iff \tau(x) < \infty.$$

Then h satisfies C. To see this, assume that there is a constraint $\chi \in C$ that is not satisfied by h. If χ is of the form $x = \wedge y_j$ with a finite number of y_j's, either $h(x) = 1$ and $h(y_j) = 0$ for some y_j, or $h(x) = 0$ and $h(y_j) = 1$ for all y_j. Neither is possible by the definition of h: if $h(y_j) = 0$ for some y_j, $\tau(y_j) = \infty$ and thus $\tau(x) = \infty$; if $\tau(y_j) < \infty$ for all y_j, then all y_j's are in X_k, where k is the maximum of $\tau(y_j)$, and thus $\tau(x) = k + 1$. If χ is of the form $x = \vee y_j$, either $h(x) = 1$ and $h(y_j) = 0$ for all y_j, or $h(x) = 0$ and $h(y_j) = 1$ for some y_j. These are not possible either: if $h(y_j) = 0$ and thus $\tau(y_j) = \infty$ for all y_j, then $\tau(x) = \infty$; if $h(y_j) = 1$ for some y_j, $\tau(y_j) < \infty$ and $\tau(x) \le \tau(y_j) + 1$.

For any x in X with $\tau(x) < \infty$, define a finite subset Y_x of $X_{\tau(x)}$ as follows:

$$x \in Y_x,$$
$$y \in Y_x, \chi_y = \text{``} y = \wedge y_j \text{''} \Rightarrow \forall y_j \in Y_x,$$
$$y \in Y_x, \chi_y = \text{``} y = \vee y_j \text{''} \Rightarrow \arg\min_{y_j, \tau(y_j) < \tau(y)} \nu(y_j) \in Y_x.$$

Since only a finite number of variables appear in any conjunction in C, for a variable y already in Y_x, each rule adds at most a finite number of variables y_j, which all satisfies $\tau(y_j) < \tau(y)$. Thus there are only a finite number of variables in Y_x. Also, if we denote $Y_x^i = Y_x \cap X_i$, each variable in Y_x^i is forced by Y_x^{i-1} for $i = 1, \cdots, \tau(x)$.

Consider the following algorithm:

45

IsOne(x)
```
 1   for each increasingly large finite subset Z ⊂ X such that x₀, x ∈ Z
 2       Y₀ ← {x₀}
 3       for i = 1, 2, ⋯
 4           Yᵢ ← Yᵢ₋₁
 5           for each z ∈ Z
 6               add z to Yᵢ if z is forced by Yᵢ₋₁
 7           until Yᵢ = Yᵢ₋₁
 8   until x ∈ Yᵢ for some i
```

On line 1, we make sure the increasing subset Z will contain each variable eventually by adding variable $v^{-1}(k)$ for $k = 0, 1, 2, \cdots$.

If $\tau(x) < \infty$, IsOne(x) terminates in finite steps, since Z will eventually include Y_x. Conversely, if IsOne(x) terminates in finite steps, the set Y_i that contains x is a subset of X_i. Thus $\tau(x) < \infty$.

Let us assume $s \in \Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid s_1)$. Since the data $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \{1\}, s_1, S)$ represents $\bar{\sigma} \subset S = \mathbb{N} \times \mathbb{N}$, any cross section s' in $\Gamma(\mathscr{S}, \mathscr{S}', \mathscr{M} \mid s_1)$ satisfies $s'(S) = \bar{\sigma}$. Thus, for any assignment g to X that satisfies C, $g(t) = g_s(t)$ for any $t \in S$. In particular, we have $h(t) = g_s(t)$ for any $t \in S$. This means that $\tau(x_{(i,b)}) < \infty$ for $(i, b) \in S$ if and only if $\sigma[i] = b$.

Now, we define a Turing machine M that executes the following algorithm:

```
 1   Given (𝒮, 𝒮', ℳ, {1}, s₁, S)
 2   Let ρ be an empty string variable
 3   for i = 0, 1, ⋯
 4       run IsOne(xₜ) in parallel for t = (i, 0), (i, 1), and (i − 1, 1 − ρ[i − 1]) if i > 0.
 5           until one of the parallel processes terminates
 6       if  i > 0 and IsOne(x₍ᵢ₋₁,₁₋ρ[i−1]₎) terminated
 7           then pop ρ[i − 1] and terminate returning ρ
 8       else if IsOne(x₍ᵢ,₀₎) terminated
 9           then ρ[i] ← 0
10       else if IsOne(x₍ᵢ,₁₎) terminated
11           then ρ[i] ← 1
```

For $i = 0, 1, \cdots, |\sigma| - 1$, IsOne($x_{(i,\sigma[i])}$) terminates in finite steps but the other processes do not. For $i = |\sigma|$, either IsOne($x_{(i,0)}$) or IsOne($x_{(i,1)}$) can terminate first. For $i = |\sigma| + 1$, only IsOne($x_{(i-1,1-\sigma[i-1])}$) terminates. Thus, M always terminates returning σ.

Finally, let $(\mathscr{S}, \mathscr{S}', \mathscr{M}, \{1\}, s_1, S)$ be the diagram that has size $I(\bar{\sigma} \mid \mathscr{M}_{\mathbb{N}})$. Since the data can be encoded in a string of length $O(I(\bar{\sigma} \mid \mathscr{M}_{\mathbb{N}}))$ and emulating M on U to run on the string produces σ, there are constants $d_U, e_U \in \mathbb{N}$ such that

$$K_U(\sigma) \le d_U I(\bar{\sigma} \mid \mathscr{M}_{\mathbb{N}}) + e_U$$

that do not depend on σ. □

By Theorem 2 and Theorem 3, we can say that $I(\bar{\sigma}|\mathcal{M}_\mathbb{N})$ is equivalent to the Kolmogorov complexity $K_U(\sigma)$. Thus, the formulation gives the larger class of objects a meaningful measure of information that generalizes Kolmogorov complexity.

7   Discussion and Conclusion

How is the information measure we defined different from Kolmogorov complexity in the general case, not the case in the previous section? Let us take the class $\mathcal{G}$ of subsets of Euclidean plane X as an example and try to follow the common prescription: that is, we fix an encoding of the objects into strings; and then we define the Kolmogorov complexity of the string encoding an object as its complexity. Consider

$$S \xrightarrow{U} S \xrightarrow{f'} \mathcal{G} \tag{23}$$

where S is the set of all strings and U is the partial map defined by a universal Turing machine. The map f' is an encoding of objects by strings. The common notion is that we should define the length of the shortest string s such that $x = f'(U(s))$ as the Kolmogorov complexity of x. However, as we mentioned in 1.1, because the cardinality of $\mathcal{G}$ is larger than that of S, we need to either encode only some of the objects, encode multiple objects by each string, or both. To allow for both possibilities, we consider

$$S \xrightarrow{U} S' \xleftarrow{f} \mathcal{G} \tag{24}$$

instead. Here, $S' = S + \{\uparrow\}$, where $f(x) = \uparrow$ means "x is not encoded." For an object x, the length of the shortest string s such that $f(x) = U(s)$ is defined as the amount $K_{f,U}(x)$ of information in x, unless $f(x) = \uparrow$, in which case $K_{f,U}(x) = \infty$.

The question is: does there exist a map f that makes $K_{f,U}(x)$ equivalent to $I(x|\mathcal{M}_E)$? The trivial answer is: yes, we can define f using $I(x|\mathcal{M}_E)$. If $I(x|\mathcal{M}_E)$ is finite, we define $f(x)$ to be the first string s (in some standard order) such that the shortest string s' with $U(s') = s$ has length $I(x|\mathcal{M}_E)$; otherwise we define $f(x) = \uparrow$. But this only highlights the meaninglessness of this kind of discussion without restricting f in some way: as we noted in the introduction, just about any "complexity" can be realized this way.

Let us instead try to define a reasonable encoding f to see if it gives rise to anything close to $I(x|\mathcal{M}_E)$. First, a standard way to encode points in X would be to identify X with $\mathbb{R}^2$ and encode the two coordinates of a point by an infinite sequence. We can also encode a countable set of points by a sequence similarly by dovetailing between more and more points, enumerating the digits for each to higher and higher precision. To accommodate this, we again modify (24) a little and consider

$$S \xrightarrow{\tilde{U}} \tilde{S} \xleftarrow{\tilde{f}} \mathcal{G} \tag{25}$$

Here, $\tilde{S}$ is the set of all infinite sequences. If a sequence s is computable, there is a Turing machine that, given a natural number n, prints the first n symbols of s and halts. Let $\tilde{U}$ be the partial map that maps an encoding of such a Turing machine to the sequence. We define the length of the shortest string s' such that $\tilde{U}(s') = \tilde{f}(x)$ as $K_{\tilde{f},\tilde{U}}(x)$. If there is no such string s', we define $K_{\tilde{f},\tilde{U}}(x) = \infty$. Note that this particular scheme is already quite different from ours. According to this scheme, the information in a single point varies in a way irrelevant to anything we may be interested in while thinking about subsets of a Euclidean plane. If the sequence corresponding to the point is uncomputable, it has infinite information. In contrast, for any point p in X, $I(\{p\}|\mathcal{M}_{\mathrm{E}}) = 1$, as we saw in section 5. We begin to see how we introduce unnecessary complications by trying to first encode everything by a string.

How about other, uncountable sets? Most of geometric objects are uncountable subsets of X. We can imagine encoding lines with a pair of points and circles with a point and a real number, etc., encoded as above. That is, we *define $\tilde{f}$* so that a line on X going through a pair of points is mapped to a sequence encoding two points and some flag indicating that it is a line. This is a definition of $\tilde{f}$ by using characteristics, or regularity, of the subset. However, the problem is that any interpretation of the sequence in terms of regularity must be incorporated in the definition of $\tilde{f}$. Thus, as we wish to add more patterns—lines, circles, repeated patterns—the definition of $\tilde{f}$ has to become more and more elaborate; and we would wish to somehow define $\tilde{f}$ computationally. However, this cannot be done because the spirit of using Kolmogorov complexity is that any information, including any regularities and redundancies, is conveyed through a sequence and any computation must be done by a Turing machine, strictly on the left side of $\tilde{S}$ in (25).

In order to define $\tilde{f}$ computationally, we would have to define the notion of computation involving $\mathcal{G}$. And, of course, that is exactly what our scheme does, where the situation is like

$$S \xrightarrow{g} \mathscr{D} \xrightarrow{h} \mathscr{G} \tag{26}$$

A given set of maps determines the possible set $\mathscr{D}$ of diagrams generated by the maps. The map h is the interpretation of diagrams into the ground representation. The difference between the two schemes is on which side of the center the computation, or the "decompression," takes place. It occurs in U in (25) and in h in (26).

Thus, it is not easy to give a reasonable and simple definition of $\tilde{f}$ that makes $K_{\tilde{f},\tilde{U}}(x)$ equivalent to $I(x|\mathcal{M}_{\mathrm{E}})$. We cannot seem to define a simple encoding $\tilde{f}$ that reflects the informal notion of information in the domain of subsets of Euclidean plane X. Even with this single, relatively simple class of objects, we have these problems. Remember that we cannot just say that there is an encoding and only talk about strings, as we pointed out repeatedly; if we insist encoding objects into strings, we need to define a concrete encoding for each class of objects.

The new representation provides a meta-definition for that purpose. It allows specifying how to embed multiple computations in larger spaces in a versatile way. As shown in the previous section, a Turing machine can be simulated by a diagram generated by $\{0, \mathrm{succ}\}$.

48

This can be isomorphically embedded in a diagram generated by $\mathscr{M}_E$ by using p and move$_v$ instead of 0 and succ, where $p \in X$ and move$_v : X \to X$ is a move by a vector v, which can be represented by a diagram. Although a computation can be in other more direct forms, as in the examples in 4.1, this shows that universal computation can be embedded as a part of the representation.

Also, it allows separating the information that we wish to ignore from that which we measure. As we have seen, this is important as some information we want to ignore is infinite. The structure maps delineate the *a priori* parts of the objects whose information are to be ignored, leaving only the information in the structure: hence the name "structural information."

Finally, the representation can provide a useful abstraction in designing real-world applications. In dealing with high-dimensional signal-level objects, such as sounds, images, and sensory readings in robots, we need to model the process of finding patterns in raw signal and grouping them together to be described at a symbol level. The information measure may be used to regularize the process of finding useful patterns in the ground level. We can imagine implementing a system using this representation, which corresponds to defining an encoding g of diagrams shown in (26).

Given a fixed set of structure maps, it can represent infinite sets by finite symbolic expressions and sparse parameters. That is, after fixing the sets and maps as primitives, diagrams can be symbolically represented just as graphs are. While the fixed sets such as Euclidean spaces and the space of real numbers have to be implemented by some system-dependant approximation such as floating point numbers, the structure itself is preserved irrespective of such approximations. In the case of the line example, although points and vectors might be translated between systems varied in, say, resolution, its structure, its "lineness," so to speak, would survive. In the current practice, an encoding of a line by such sparse data would have to be written into the code; leading to the lack of generality and flexibility.

Also, diagrams can be naturally understood as modules to construct larger and more complex ones. The set union and intersection are the most basic ways to combine them; or we can put the output of one diagram into the input of another. Ultimately, any computation can be used to combine them. Our formulation provides a useful abstraction that allows us to model such structures in a uniform way so that they can be manipulated automatically.

Conclusion

In this paper, we introduced a uniform representation of general objects. In an abstraction of the dense representation, which includes strings, bitmaps, and other raw data, the objects are assumed to be given *a priori* as subsets of some sets. The proposed representation uses a new construct called *diagram* to represent objects with regularity through sparse parameters, using the maps that characterize the space in which the objects are included as subsets. Since the representation can emulate any computation, it can exploit any computable regularities in objects to compactly describe them. It is also general enough to represent random objects as raw data, making it possible to interpolate between the dense

and sparse representation. There is a prescribed way to connect the description to the raw data; thus, the representation is *grounded*. In other words, the relationship between the parameters and the data is part of the representation so that we can give raw data and then ask what sparser, more structured representation is possible. With the representation, we also defined a measure of information in the objects. We proved that the measure is equivalent to Kolmogorov complexity in the case of strings.

To answer our question in the introduction, we would say that a subset A of a set X with structures characterized by a set $\mathcal{M}$ of maps is a pattern if $I(A\,|\,\mathcal{M})$ is significantly smaller than $|A|$ in the case A is finite and, in the case A is infinite, if $I(A\,|\,\mathcal{M})$ is finite. Even when $I(A\,|\,\mathcal{M})$ is infinite, there can be a pattern in a statistical sense, which we leave for future work.

References

[1] G. J. Chaitin. On the Length of Programs for Computing Finite Binary Sequences. *J. ACM* **13**:547–569, 1966.

[2] G. J. Chaitin. On the Length of Programs for Computing Finite Binary Sequences: Statistical Considerations. *J. ACM* **16**:145–159, 1969.

[3] B. Edmonds. Syntactic Measures of Complexity. Doctoral Thesis (also CPM Report No.: 99-55), University of Manchester, Manchester, UK. 1999.

[4] K. S. Fu. *Syntactic Methods in Pattern Recognition*. Academic Press, 1974.

[5] R. L. Gregory, *The Intelligent Eye*, Weidenfeld & Nicolson, London, 1970.

[6] U. Grenander. *Lectures in Pattern Theory*, volumes I-III. Springer-Verlag, 1976, 1978, 1981.

[7] U. Grenander. *General Pattern Theory*. Oxford University Press, 1993.

[8] U. Grenander and M. Miller. *Pattern Theory*. Oxford University Press, 2007.

[9] A. N. Kolmogorov. Three Approaches to the Quantitative Definition of Information. *Problems of Information Transmission* **1**:4–7, 1965.

[10] M. Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications (2nd ed.)* Springer-Verlag, 1997.

[11] J. Rissanen. Modeling by the shortest data description. *Automatica* **14**:465–471, 1978.

[12] C. E. Shannon. The mathematical theory of communication. *Bell System Tech. J.*, **27**:379–423, 623–656, 1948.

[13] R. J. Solomonoff. A Preliminary Report on a General Theory of Inductive Inference. Report ZTB-135, Zator Co., Cambridge, MA, 1960.

[14] R. J. Solomonoff. A Formal Theory of Inductive Inference. *Information and Control*, **7**:1–22, 224–254, 1964.

[15] A. P. Witkin and J. M. Tenenbaum. On the Role of Structure in Vision. In *Human and Machine Vision*, pp. 481–543. Academic Press, New York, 1983.